4
NR INDEKSU 353965
PL ISSN 0860-1674

Z MIKROKOMPUTEREM NA TY

Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH NR 4 (28) KWIECIEŃ 1988 CENA 150 ZŁ

MECENAS

DRUKARKA CITIZEN 120 D

JOYTEST

ELEKTRONIKA JOYSTICKA

ALGORYTM HORNERA

AUTOSTART PROGRAMÓW

GUN FRIGHT

W JACHRANCE O BAJTKU

Ani się obejrzeliśmy, a „Bajtek" wszedł w trzeci rok swego rozwoju. Zaczynaliśmy jako pionierzy — obecnie przybyło nam towarzystwo licznych pism i wydawnictw na tematy związane z informatyką. Zmieniły się w tym czasie realia rynku komputerowego w Polsce i uwarunkowania rozwojowe tej kluczowej skądinąd dziedziny współczesnej gospodarki. Musimy więc również zmienić się i my.

Przez dwa dni — od piątkowego popołudnia 18 marca poczynając — trwało w Jachrance spotkanie programowe dziennikarzy i współpracowników „Bajtka". Bez taryfy ulgowej podsumowaliśmy dotychczasowe publikacje i działania naszego pisma. Nie ukrywamy, że bilans wypadł pozytywnie, choć i lista grzechów okazała się długa. Nie miejsce tu na szczegółowe ich relacjonowanie; najważniejsze dla Czytelnika jest przecież to, co będzie mógł spotkać w przyszłości na łamach „Bajtka". I temu właśnie poświęcona była większość naszych dyskusji w Jachrance.

Bylibyśmy zarozumiali uważając, że potrafimy wszystkie problemy rozstrzygnąć w redakcyjnym gronie. Dlatego do naszej dyskusji zaprosiliśmy wypróbowanych sojuszników „Bajtka".

Długo pozostanie w pamięci wszystkich „bajtkowców" wielogodzinne spotkanie z profesorem Władysławem M. Turskim poświęcone, najogólniej mówiąc, współczesnym trendom rozwoju informatyki. Umówiliśmy się wcześniej z Profesorem na formułę spotkania wewnętrznego, nie do publikacji więc, dotrzymując zobowiązania, nie będę składał relacji z dyskusji. Mogę jednak przekazać uwagę jednego z naszych kolegów wypowiedzianą po spotkaniu: „Wystarczy to nagrać i opublikować — a byłby bestseller!" W imieniu zespołu „Bajtka" zachęcam tą drogą Pana Profesora do napisania kolejnego hitu na wzór jego książki pt. „Nie samą informatyką". To się sprzeda! A co ważniejsze — na to czeka młodzież!

Z profesorem Turskim dyskutowaliśmy między innymi o granicach programowania; natomiast granice technologii były wiodącym tematem spotkania z doc. dr hab. inż. Andrzejem Jakubowskim, kierownikiem Zakładu Mikroelektroniki na Wydziale Elektroniki Politechniki Warszawskiej. Najbardziej łapidarny wniosek z tego spotkania: w XXI wieku wejdziemy jeszcze z technologią opartą na krzemie. Co będzie potem — to na razie temat na fascynującą powieść science-fiction, ale tylko na razie...

Cywilizacyjne konsekwencje rewolucji informacyjnej przedstawił nam doc. dr hab. Lech Zachor z Uniwersytetu im. Marii Curie-Skłodowskiej w Lublinie. Wrócimy do tego tematu na łamach „Bajtka". Komputeryzacja nie jest przecież celem w sobie. Komputery dokonują rewolucji, a więc sterując rozwojem komputeryzacji można wpływać na przyspieszenie lub zwolnienie premier cywilizacyjnych. Na ile i pod jakimi warunkami jest to możliwe?

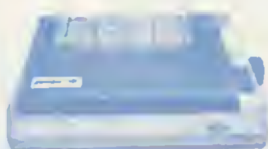
O problemach zapewnienia dostaw taniego sprzętu mikrokomputerowego dyskutowaliśmy z dyrektorem Wojciechem Szanterem z Centralnej Składnicy Harcerskiej. W meandry wprowadzania informatyki do szkół wprowadzał nas profesor Kuran z warszawskiego LO im. Czackiego. Pani Irena Danda z Fundacji „Pomoc dzieciom" zapoznała nas z dotychczasowymi działaniami i doświadczeniami w zakresie wykorzystania komputerów do pomocy dzieciom chorym...

Każde z tych spotkań zaowocuje na łamach „Bajtka" różnorodnymi publikacjami i akcjami.

Oczywiście, odbyły się również „burze mózgów" i dyskusje na tematy dotyczące bezpośrednio naszego pisma. Również z udziałem przyjaciół „Bajtka" — redaktora Jana Rurańskiego, znanego popularyzatora nauki i techniki oraz dyrektora Wiesława Fałczaka z Młodzieżowej Agencji Wydawniczej.

Spędziliśmy te dwa dni z Jachrance bardzo pracowicie. Na ile owocna była ta praca? To Czytelnicy będą mogli ocenić sami podczas lektury kolejnych numerów naszego pisma oraz wydań specjalnych „Bajtka" adresowanych do użytkowników poszczególnych typów komputerów.

Waldemar Siwiński



SZANOWNY PANIE REDAKTORZE!

Prenumeruję wasze pismo, pracuję z nim. Pracuje z „Bajtkiem" także wielu moich kolegów i bardzo się nam on podoba. Szkoda, że nie można go było w naszym kraju wcześniej prenumerować, dobrze że doszedł do nas chociaż w 1988 roku.

W Związku Radzieckim pism o podobnym profilu niestety nie ma, chociaż zainteresowanie tą tematyką jest ogromne. Dlatego też „Bajtek" jest dla nas tak aktualny.

Ja i moi przyjaciele mamy do Was wielką prośbę. Jeśli tylko macie taką możliwość przyślijcie nam pełny komplet egzemplarzy waszego pisma za 1986 i 1987 rok.

Mamy wielką nadzieję, że nam pomożecie.

Władimir Korbut
Kijów

Od red. *Zgodnie z ogłoszeniem zamieszczonym w nr 2/88 będziemy kontynuowali wysyłkę (za zaliczeniem pocztowym) zaległych numerów „Bajtka". Niestety nie wszystkie mamy do dyspozycji. Daremne pozostaną dla przykładu prośby o „Bajtka" „Tylko o Atari".*

WYBIERZ SAM

„BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH"

ADRES: 00-687 Warszawa, ul. Wspólna 61 Tel. 21-12-05

Przewodniczący Rady Redakcyjnej: Jerzy Domański-redaktor naczelny „Sztandaru Młodych".

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM" — kierownik zespołu „Bajtka"), Grzegorz Onichimowski (sekretarz redakcji „Bajtka"), Roman Poznański (kierownik działu klanów), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Pilaszek, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Walgórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski

Klany redagują:
Commodore — Klaudiusz Dybowski, Dominik Falkowski
Amstrad-Schneider — Jonasz Mayer
Spectrum — Marcin Przasnyski
Atari — Wojciech Zientara, Sergiusz Piotrowski

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wóltańska

WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49 Redakcja Reklamy 13-20-40 do 49 w. 403, 414
Cena 100 zł.
Skład techniką CRT-200, przygotowalnia offsetowa i druk PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51
Nr zlecenia 010188 n. 150 000 egz. U-113



Bajtek

Rozmowa z **Mirosławem Madejskim,** przewodniczącym Rady Fundatorów fundacji „Pomoc Dzieciom", szefem „Refleksu"

ŚMIEJĄ SIĘ Z NAS CZASEM, ŻE INWESTUJEMY W TAK DALEKĄ PRZYSZŁOŚĆ



MECENAS

— Panie prezesie, skąd wziął się pomysł zorganizowania przez właścicieli prywatnych przedsiębiorstw branży komputerowej instytucji charytatywnej. Czyżby z niechęci do płacenia podatków?

— Forma uczestnictwa w działalności charytatywnej jaką jest fundacja przyjęła się dość powszechnie na świecie. W wielu krajach, takich jak np. Stany Zjednoczone państwo, w ogóle, w niewielkim stopniu zajmuje się zabezpieczaniem potrzeb społecznych obywateli. Rolę tę przejmują właśnie rozliczne istytucje powołane przez osoby prywatne, spółki, konsorcja. U nas również Sejm dostrzegł potrzebę tworzenia fundacji. Uchwaleniu ustawy o fundacjach nie towarzyszyły jednak żadne zmiany w systemie podatkowym. Tak więc łożąc na cel społeczny nie płacimy bynajmniej mniejszych podatków.

— A zatem — co kierowało fundatorami?

— Myślę, że nade wszystko zamiłowanie do informatyki i właśnie chęć fundacji dzieciom. Przecież niezależnie od powołania samej fundacji, także w przeszłości „Refleks" pomagał wyposażać pracownie informatyczne w sprzęt komputerowy, oddawał swoje laboratoria na zajęcia dla dzieci. A ludzie u nas pracujący za darmo zostawali wówczas po godzinach, by z nimi pracować. Śmieją się z nas czasem, że inwestujemy w tak odległą przyszłość.

Z drugiej strony szkoda nam było trochę rezygnować z produkcji tego wszystkiego, co już nie jest na rynku całkowitą nowością, lecz mogłoby się jeszcze z powodzeniem sprzedawać. W ofercie samego „Refleksu" na produkty te nie ma już miejsca, fundacja dla wsparcia swej działalności charytatywnej mogłaby nimi handlować.

— A komu konkretnie będzie pomagać fundacja?

— Nie mamy ambicji stworzenia jakiejś [illegible] instytucji. Chcemy [illegible] tylko dwiema grupami dzieci [illegible] będą z dzieci [illegible] sprawnych ruchowo [illegible] tyl[illegible] do [illegible] w [illegible] grafików [illegible] oni [illegible] zawód [illegible]

— Wspomniał pan o dwóch grupach. Kto znajdzie się w tej drugiej?

— [illegible] wybitnych zdolno[illegible]

— Jakie widzi pan przed nimi zadania?

— Chcielibyśmy, by niejako „ciągnęli" tę pierwszą grupę. Poza tym, wydaje mi się, że mogliby brać czynny udział w opracowywaniu oprogramowania edukacyjnego, a w każdym razie je testować.

— Pisanie programów edukacyjnych, to niełatwe zadanie.

— Dlatego powierzyć je chcemy najlepszym. A że będą to tak młodzi ludzie? Na [illegible] problemy przy opracowywaniu programów edukacyjnych [illegible] do w[illegible] [illegible] zadań w ich obe[illegible]

— Skąd fundacja znajdzie fundusze na swą działalność, gdzie znajdzie dla niej bazę lokalową i sprzęt?

— Jak na razie [illegible] się na [illegible] fundatorów. W [illegible] w dziedzinach handlu, reklamy, informatyki, produkcji z tworzyw sztucznych. Mam nadzieję, że dochody z tej działalności pozwolą w całości sfinansować zajęcia z obiema wspomnianymi grupami. Planujemy nawet osiągnięcie pewnych nadwyżek, które przekazać chcielibyśmy m.in. fundacji „Złoty kłos" zajmującej się pomocą dzieciom chorym na celiakię.

Sprzęt i lokale, jak na razie, udostępnił fundacji „Refleks". W trakcie trwania wystawy KOMPUTER 88 [illegible] się [illegible] wy[illegible] tową. Żadna z polskich firm [illegible] [illegible]

— A jak wyglądać będzie dobór uczestników obu wspieranych przez fundację grup.

— [illegible] w [illegible] Natomiast [illegible] dokonywać będziemy na [illegible] Liczymy m.in. na zgłaszanie nam kandydatów przez ich nauczycieli bądź rodziców. Myślę, że znaczną część spośród naszych przyszłych podopiecznych stanowić będą Czytelnicy „Bajtka". Dlatego też liczę na to, że dzięki informacji o powstaniu i działalności naszej fundacji pojawią się zgłoszenia od tych, którzy liczyliby na jej pomoc.

— W takim razie proponujemy przysługa za przysługę. Gdy podopieczni fundacji na dobre rozwiną już swoją twórczość programistów mogliby na naszych łamach upowszechniać swoje dzieła

— Oczywiście [illegible] Myślę [illegible] dobrze byłoby gdyby w [illegible] w „B[illegible] [illegible] 14-15 letni informatycy [illegible] Byłoby [illegible] że zwracaliby [illegible] To byłyby [illegible] tworzenie [illegible] prawdę [illegible]

— Oczekujemy więc w „Bajtku" na pisemne, telefoniczne lub osobiste zgłoszenia wszystkich zainteresowanych propozycją Fundacji. Panu dziękujemy natomiast za rozmowę.

Rozmawiali:
Grzegorz Onichimowski
Kazimierz Treger

TEST !

Drukarkę Citizen 120-D otrzymaliśmy do testowania dzięki uprzejmości firmy „Synelec".

DRUKARKA

Citizen 120-D to 9-igłowa półprofesjonalna drukarka mozaikowa przeznaczona do współpracy zarówno z domowymi komputerami 8-bitowymi (Amstrad, Atari, Commodore), jak i komputerami klasy IBM PC.

BUDOWA I DZIAŁANIE

Konstrukcja mechaniczna drukarki jest klasyczna i nie odbiega od innych urządzeń tej klasy. Możliwe jest użycie papieru perforowanego, z rolki i w arkuszach o szerokości od 76 do 250 mm. Maksymalnie można uzyskać oryginał i dwie kopie. Dodatkowym wyposażeniem jest podajnik papieru perforowanego umieszczany ponad wałkiem i połączony z mechanizmem drukarki przez kółko zębate. W przypadku zastosowania ażurowego stojaka możliwe jest podawanie papieru spod drukarki przez szczelinę w spodniej części obudowy.

W bardzo interesujący sposób rozwiązano część elektroniczną drukarki. Wszystkie układy oprócz bezpośrednio sterujących pracą części mechanicznej zostały umieszczone na oddzielnej płytce wysuwanej jak szuflada z prawej strony obudowy. Znajduje się tam również gniazdo złącza Centronics.

Głowica drukarki umieszczona została przed wałkiem na stalowej prowadnicy i przesuwana jest paskiem zębatym. Demontaż głowicy jest bardzo prosty. Taśma drukująca o szerokości 6 mm znajduje się w dużej kasecie, a głowica przesuwa się między taśmą a kasetą. Prędkość druku wynosi w trybie draft 120 CPS (znaków na sekundę), a w trybie korespondencyjnym (NLQ) — 24 znaki na sekundę, jest więc nieco mniejsza niż w innych drukarkach.

Znaki drukowane są z matrycy 9 na 9 punktów w trybie draft oraz 17 na 17 w trybie NLQ. Do dyspozycji są cztery podstawowe kroje pisma: Pica — 10 CPI (znaków na cal), Elite — 12 CPI, Condensed Pica — 17 CPI i Condensed Elite — 20 CPI. Odpowiednio uzyskuje się 80, 96, 136 i 160 znaków w wierszu (na arkuszu A4). Dodatkowo można wykorzystać druk proporcjonalny, w którym liczba znaków zależy od ich rodzaju ("i" zajmuje mniej miejsca niż np. "k").

Wszystkie kroje pisma mogą być pochylone (italics) lub drukowane w podwójnej szerokości (5, 6, 8,5 i 10 CPI). Dodatkowo istnieją dwie możliwości druku rzadko spotykane w tej klasie sprzętu: druk odwrócony (w negatywie) i druk powiększony pionowo (znaki o podwójnej wysokości). Różne gęstości pisma pokazuje rys. 1, a różne odmiany kroju rys. 3. Przez odpowiedni wybór dodatkowych opcji można uzyskać różne warianty wyrazistości druku (rys. 2).

Wszystkie drukarki posiadają już standardowo odstęp wierszy 1/6, 1/8 i 7/72 cala. Poza tym istnieje możliwość dowolnego ustalenia odstępu z określonym krokiem. Większość firm stosuje programowany odstęp n/216 cala, a firma Star n/144 cala. I tu Citizen ma przewagę, bowiem dysponuje obydwoma wariantami.

Szczególnie bogate są możliwości Citizena w operowaniu odstępami i tabulacjami. Mamy do dyspozycji poziomą tabulację stałą, zmienną i względną oraz względną i bezwzględną tabulację punktową (określaną w punktach, a nie w znakach). Trzy pierwsze rodzaje dotyczą także tabulacji pionowej. Kolejną funkcją jest możliwość ustalania odstępu między znakami w zakresie od 0 do 127 punktów (rys. 4). Przy druku korespondencyjnym możliwe jest justowanie tekstu przez samą drukarkę (nie przez komputer). Do wyboru mamy centrowanie tekstu, justowanie do lewego lub prawego marginesu oraz justowanie do obu marginesów.

```
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
```

Rys. 1

Drukarka może pracować w standardach Epson FX i IBM Graphics Printer. Standard wybierany jest przełącznikiem lub programowo — jest to pierwsza ze znanych mi drukarek posiadająca możliwość przełączenia programowego. W trybie Epson do dyspozycji jest 96 znaków normalnych lub pochyłonych oraz 11 zestawów międzynarodowych po 12 znaków w każdym. Poza tym możliwe jest uzyskanie zamiast kodów sterujących od 128 do 159 dodatkowych 32 znaków graficznych. W trybie IBM otrzymuje się pełny zestaw liczący 232 znaki.

W obu trybach pracy (Epson i IBM) możliwe jest projektowanie własnych znaków i zapamiętanie ich przez drukarkę. Niestety zmniejsza to szybkość współpracy z komputerem, ponieważ pojemność pamięci drukarki wynosi tylko 4 KB (większość drukarek posiada bufor 8 KB).

Do sterowania pracą drukarki służą przyciski znajdujące się na górnej pokrywie obudowy. Służą one do połączenia i odłączenia drukarki (ON LINE), przesunięcia o stronę (FORM FEED) i przesunięcia o wiersz (LINE FEED). Oprócz tego kombinowane użycie przycisków pozwala na wybranie jednego z krojów pisma: Pica, NLQ, Italics, Emphasized lub Reduced. Niestety wymaga to pewnej wprawy. Obok przycisków znajdują się diody sygnalizujące włączenie zasilania, gotowość do pracy i brak papieru.

Włączenie drukarki z wciśniętymi przyciskami FF i LF wywołuje tryb ko-



Inne drukarki „Citizen" oferowane na nasz rynek przez firmę „Synelec" to: MSP-10E, MSP-15E, MSP-40, MSP-45, MSP-50 (kolor), MSP-55 (kolor), HQP-45 (24-igłowa), Premiere 35 (rozetkowa).

W jednym z najbliższych numerów „Bajtka" zamieścimy test kolejnej drukarki CITIZEN.

CITIZEN 120 D

Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D

Rys. 2

Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D

Rys. 3

piowania szesnastkowego. W odróżnieniu od poprzednio testowanych drukarek, oprócz kodów drukowane są także odpowiadające im znaki ASCII. Wciśnięcie przycisku LF przy włączaniu wywołuje, jak w innych urządzeniach, test drukarki — drukowane są wszystkie znaki dostępne w wybranym trybie pracy. Dodatkowo po włączeniu z wciśniętymi przyciskami LF i ON LINE drukowany jest numer zastosowanego w drukarce układu ROM i EPROM z zestawem znaków.

Do stałego ustawienia parametrów pracy drukarki służy zespół 10 przełączników umieszczonych na płytce z elektroniką. Dostępne są one po wysunięciu tej płytki lub przez niewielkie okienko obok prowadnicy głowicy drukującej.

EKSPLOATACJA

Podczas testowania badana była współpraca drukarki z komputerami Amstrad, Atari i Commodore. We wszystkich przypadkach drukarka pracowała bez zarzutu. Jakość wydruków jest zbliżona do uzyskiwanych na poprzednio testowanej drukarce DMP3160. Również tu w trybie graficznym pozostawał ślad przejścia głowicy.

Użytkowanie drukarki jest proste i wygodne. Jedynie wybieranie krojów pisma przy pomocy przycisków wymaga dokładności i dobrej pamięci (lub zajrzenia do instrukcji). Bardzo dogodna jest natomiast możliwość programowego przełączania trybów Epson i IBM. Przy korzystaniu z papieru w arkuszach daje się odczuć brak automatycznego podajnika — znajdujący się na wyposażeniu drukarki służy jedynie do papieru z perforacją. Z drugiej strony brak ostrej krawędzi ułatwiającej oderwanie papieru z rolki.

Kaseta zawiera dużą ilość taśmy drukującej, co zapewnia duży odstęp pomiędzy jej kolejnymi renowacjami. Dostęp do kasety i głowicy jest bardzo prosty i ich demontaż może wykonać każdy (oczywiście w przypadku głowicy należy korzystać z instrukcji).

Praca drukarki jest stosunkowo cicha pomimo braku bezpośredniej osłony głowicy. Poziom hałasu jest zbliżony do wytwarzanego przez inne drukarki tej klasy.

Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D
Drukarka Citizen 120-D

Rys. 4

PODSUMOWANIE

Drukarka Citizen 120-D wyróżnia się w swojej klasie cenowej przede wszystkim znacznie większymi możliwościami wydruku tekstów. Więcej jest dostępnych odmian druku i ich kombinacji. Posiada ona także więcej znaków w trybie Epson oraz dodatkowy, ósmy tryb graficzny. Duża żywotność taśmy predestynuje ją dla użytkowników wykonujących duże ilości wydruków jednorazowo.

Zalety drukarki:
- duża liczba zestawów znaków w trybie Epson
- dodatkowe znaki graficzne w trybie Epson
- duża liczba krojów pisma i ich kombinacji
- programowe przełączanie trybów Epson i IBM
- możliwość justowania wydruku
- dużo wariantów tabulacji i formatowania wydruku
- pełny wydruk przy kopiowaniu szesnastkowym

Wady drukarki:
- skomplikowany sposób wyboru krojów pisma przyciskami
- brak krawędzi do odrywania części papieru z rolki
- mała pojemność bufora

Wojciech Zientara

PARAMETRY TECHNICZNE DRUKARKI CITIZEN 120-D WG SPECYFIKACJI PRODUCENTA:

głowica	— 9-igłowa		n/144 lub n/216 cala
szybkość druku	— 120 CPS w trybie draft, 24 CPS w trybie NLQ	papier	— szerokość 3—10 cali grubość 0,06—0,3 mm
matryca znaków	— 9 na 9 punktów (draft), 17 na 17 punktów (NLQ)	trwałość taśmy	— 2 mln znaków
		trwałość głowicy	— 10 mln znaków
wielkość znaków	— 2,4 na 2,4 mm	interfejs	— równoległy Centronics 8-bitowy (Amphenol)
gęstość druku	— 20, 17, 15, 13,3, 12, 10, 8,5, 6 i 5 znaków na cal	zasilanie	— 180—264 V, 46—60 Hz, 100 VA
gęstość grafiki	— 60, 72, 80, 90, 120, 144 i 240 punktów na cal	wymiary	— długość 370 mm szerokość 238 mm wysokość 90 mm
odstęp wierszy	— 1/6, 1/8, 7/72, n/72,	masa	— 3,7 kg

OD ŚRODKA cz. IV

Ostatnio poznaliśmy sposób, w jaki przechowywana jest w pamięci każda linia BASIC-a, oraz co oznaczają takie znaki kontrolne, jak CHR$ 14 i CHR$ 8. W tym odcinku poznamy pozostałe znaki kontrolne. Trzy z nich dotyczą zmiany miejsca wydruku, sześć — zmiany atrybutów.

Pierwszym z nich jest CHR$ 6 — „COMMA CONTROL", czyli przecinek kontrolny. Ma on takie samo działanie, jak przecinek separujący teksty w instrukcji PRINT, tzn. drukuje tyle spacji (ale zawsze co najmniej jedną), by znaleźć się w kolumnie 0 lub 16:

PRINT "1", "2"

oraz PRINT "1"+CHR$ 6+"2"

m[illegible]ne z[illegible].

Podobnie znak CHR$ 22 — „AT CTRL", czyli AT kontrolny, pozwala przenosić pozycję wydruku w dowolne miejsce ekranu tak jak AT w instrukcji PRINT. Po tym znaku muszą wystąpić dwa bajty, określające numer linii i numer kolumny, w której ma zostać umieszczony następny znak

PRINT AT 10,7, "!"

jest równoznaczne z

PRINT CHR$ 22;CHR$ 10:CHR$ 7;"!".

Aby się przekonać, jak za pomocą tego znaku uniemożliwić prawidłowy listing programu, wpisz np.:

```
10 RANDOMIZE USR 30000: REM ...Nic nie widać! ..
```

— po instrukcji REM wpisz trzy spacje, a po wykrzykniku — dwa przecinki kontrolne. Można je uzyskać bezpośrednio z klawiatury, wciskając kolejno klawisze: EXTEND (lub dwa SHIFT-y na raz) by uzyskać kursor „E", a następnie klawisz „6" (kursor zmieni kolor na żółty) i DELETE — kursor przeskoczy do najbliższej połówki ekranu.

Po wpisaniu tej linii, wymieniamy te trzy spacje na znak AT 0,0 przez

POKE 23774,22:POKE 23775,0: POKE 23776,0.

Spróbujmy teraz wylistować program. Na ekranie nie pojawia się tekst całej linii — początkowa jej część jest zakrywana przez napis znajdujący się za instrukcją REM i znakiem AT CTRL. Podobne kłopoty są gdy ściągniemy tę linię do edycji (klawisz EDIT).

Podane współrzędne w znaku AT CTRL powinny mieścić się na ekranie, tzn. numer linii nie może być większy niż 21, a numer kolumny — niż 31. Podanie większych wartości w przypadku PRINT lub LIST powoduje komunikat „Out of screen" i zaprzestanie dalszego drukowania. Jeśli zaś listing uzyskano przez wciśnięcie ENTER (automatyczny listing) — także nastąpi zakończenie listowania programu, a ponadto na dole ekranu znajdzie się migający znak zapytania — sygnał błędu. Jest to więc praktyczny sposób na uniemożliwienie wylistowania każdego programu.

Kolejnym znakiem kontrolnym jest CHR$ 23 — „TAB CTRL" — znak tabulacji poziomej. Po nim następują dwa bajty określające numer kolumny, do której ma zostać przeniesiona pozycja wydruku. Są one traktowane jako jedna, dwubajtowa liczba (pierwszy bajt mniej znaczący). Ponieważ są jednak tylko 32 kolumny, to jest ona brana modulo 32, czyli jej starszy bajt i trzy najstarsze bity młodszego bajtu są ignorowane. Drugą istotną sprawą jest to, że TAB przenosi pozycję wydruku przez drukowanie spacji — podobnie jak przecinek kontrolny, może być więc użyty do zakrywania znajdujących się już na ekranie tekstów.

Następną grupę znaków kontrolnych stanowią znaki zmieniające atrybuty. Są to:

CHR$ 16 — INK CTRL
CHR$ 17 — PAPER CTRL
CHR$ 18 — FLASH CTRL
CHR$ 19 — BRIGHT CTRL
CHR$ 20 — INVERSE CTRL
i CHR$ 21 — OVER CTRL.

Po każdym z tych znaków konieczny jest jeden bajt, precyzujący o jaki atrybut chodzi. Po znakach INK i PAPER mogą to być liczby 0 ... 9, po FLASH i BRIGHT: 0,1 i 8, po INVERSE i OVER : 0 i 1. Podanie innych wartości wywołuje komunikat „Invalid colour" i oczywiście przerwanie listowania programu.

Odśmiecając program zabezpieczony znakami kolorów kontrolnych ustalamy sobie np., że jeśli przeglądając treść programu napotkamy kod znaku PAPER CTRL, to wpisujemy w jego drugim bajcie wartość 0, jeśli INK CTRL — wartość 7, a w pozostałe znaki kontrolne kolorów — wartość 0. Poza tym usuwamy wszystkie przeszkadzające znaki BACKSPACE (CHR$ 8) przez zastąpienie ich spacjami (CHR$ 32). Podobnie likwidujemy znaki AT CTRL — wymieniamy za pomocą POKE-ów trzy bajty znaku na spację. Po takiej korekcie program daje się już wylistować bez żadnych niespodzianek.

Jeśli chcesz się włamać do programu ładującego, nie musisz, a w zasadzie nie powinieneś go odśmiecać — ważne jest przecież to, aby dowiedzieć się co ten program robi, w jaki sposób ładuje do pamięci i uruchamia następne bloki, a nie aby robił to „ładnie" i był napisany czysto i przejrzyście. Jest to ważne tym bardziej, że zanim nie poznasz dokładnie programu, lepiej nie rób w nim żadnych zmian — jedno zabezpieczenie może być sprawdzane przez inne. Dlatego najlepszym sposobem złamania programu jest analiza jego działania krok po kroku odczytując kolejne bajty pamięci:

```
[illegible] BORDER 0: PAPER 0: INK 0: CLS : PRI
NT #0,"LOADING";: FOR n=0 TO 20 STEP 4:
BEEP .2,n: NEXT n: LOAD ""CODE : PRINT A
T 19,0;: LOAD ""CODE : PRINT AT 19,0;: L
OAD ""CODE : PRINT AT 19,0;: LOAD ""CODE
 : PRINT AT 19,0;: LOAD ""CODE : RANDOMI
ZE USR 24064
```

Pamiętaj o właściwej interpretacji kolejnych bajtów: najpierw dwa bajty numeru linii, potem dwa bajty oznaczające jej długość (które mogą być fałszywe), a następnie tekst: instrukcja BASIC-a, potem jej parametry, za każdą liczbą CHR$ 14 i pięć bajtów zawierających wartość tej liczby. Za parametrami — dwukropek i następna instrukcja lub ENTER i nowa linia programu.

To by było na tyle, jeśli chodzi o znaki kontrolne. Jest jednak jeszcze jedna rzecz, którą trzeba wyjaśnić, abyś nie miał kłopotów z odczytywaniem BASIC-a. Chodzi o instrukcję DEF FN. Wpisz, a następnie obejrzyj dokładnie taką linię:

10 DEF FN a(a,b$,c)=a+c

Wydaje się, że powinna ona zająć w pamięci 19 bajtów (numer linii, jej długość, ENTER oraz 14 wpisanych znaków), ale tak nie jest. Interpreter po każdym parametrze funkcji umieścił znak CHR$ 14 i zarezerwował na coś następne pięć bajtów. Na co? — zaraz się przekonasz. Wpisz:

PRINT FN a (1,"123",2)

i ponownie dokładnie obejrzyj zawartość pamięci od adresu 23755. Po pierwszym parametrze w definicji funkcji dalej znajduje się CHR$ 14, ale po nim znalazły się kolejno: 0, 0, 1, 0, 0 co w pięciobajtowym zapisie oznacza liczbę 1. Podobnie, po trzecim parametrze funkcji znajduje się CHR$ 14 i bajty oznaczające liczbę 2. Po parametrze b$ także znalazła się wartość użytego parametru: CHR$ 14 i pięć bajtów, które kolejno oznaczają: pierwszy nie jest dla nas ważny, drugi i trzeci to adres, pod którym znajdował się łańcuch „123" (wywołanie funkcji nastąpiło w trybie bezpośrednim, więc adres ten dotyczy obszaru edycji linii BASIC-a), a bajty czwarty i piąty to długość łańcucha — w naszym przypadku wynosi ona 3 znaki.

Pamiętaj o tym odczytując BASIC przez PEEK a nie LIST. Zdarza się czasem, że w tych właśnie bajtach zarezerwowanych na rzeczywiste wartości funkcji ukryte są zabezpieczenia warunkujące działanie programu lub nawet program maszynowy ładujący następne bloki (np. Beta, Basic 1.0).

Na zakończenie parę słów o programach ładujących — tzw. loaderach. Ich zadaniem jest wczytanie i uruchomienie wszystkich bloków składających się na program. Zazwyczaj robią to w sposób maksymalnie utrudniający zrozumienie ich działania — tak, aby uruchomienie programu w inny sposób niż przez tego ładowacza (czyli w praktyce — włamanie się do niego) było niemożliwe. Robią to w mniej lub bardziej wyrafinowany sposób. Spójrzmy na ładowacze stosowane w większości produktów firmy ULTIMATE (np. Atic Atac, Knightlore, Pentagram, Night Shade itp.). Wyglądają one mniej więcej tak:

```
FOR n=23755 TO PEEK 23627+256*PEEK 23628
: PRINT n:"    ";PEEK n,CHR$ PEEK n AND P
EEK n>31: NEXT n
```

Za takim ładowaczem znajduje się na taśmie pięć następnych bloków: ekran oraz zakodowany główny blok programu, a za nimi trzy króciutkie bloki zabezpieczające program: jednobajtowy (kod instrukcji JP (HL)), kilkunastobajtowy (jest to procedura odkodowująca cały program), oraz ostatni — dwubajtowy, ładowany pod adres 23672, czyli do zmiennej FRAMES. Zawartość tej zmiennej jest zwiększona o 1 co 1/50 sekundy. W programie maszynowym uruchomionym przez RANDOMIZE USR 24064 wartość ta jest sprawdzana i jeśli jest inna, niż być powinna (co oznacza, że w międzyczasie, już po załadowaniu, program był zatrzymany na jakiś czas), następuje wyzerowanie komputera. Włamanie się do tego typu programów jest proste. Wystarczy załadować wszystkie bloki za wyjątkiem ostatniego, a po wylistowaniu programu, czy zrobieniu w nim odpowiednich zmian (np. wpisaniu POKE-ów unieśmiertelniających grę) wystarczy tylko wpisać LOAD "" CODE: RANDOMIZE USR 24064 (ale koniecznie w jednej linii, rozdzielając instrukcje dwukropkiem), by uruchomić grę.

Oddzielną sprawą jest odkodowanie programu czyli uruchomienie procedurki odkodowującej w taki sposób, by po zakończeniu działania wróciła do BASIC-a. Czytelnikom znającym asembler nie powinno to sprawić kłopotu, jednak ze względu na powszechność stosowania tego typu zabezpieczeń [illegible] w programach użytkowych (np. Art Studio czy The Last Word), wrócimy jeszcze do tego tematu.

A [illegible] zakończymy sprawę programów ładujących, tzn. ich BASIC-owej części i już na dobre zajmiemy się asemblerem.

Tomasz Surmacz
Robert Dudzik

Z KLAWIATURY NA JOYSTICK (1)

Chcąc zapewnić sobie możliwość długotrwałego użytkowania ZX SPECTRUM, musimy stale pamiętać o tym, aby oprócz przestrzegania podstawowych wytycznych instrukcji obsługi dbać o stan klawiatury.

Mechaniczne wycieranie się folii i uszkodzenia ścieżek przewodzących znajdujących się pod klawiszami wymienia się jako jedną z głównych wad, która niestety nie została całkowicie usunięta w nowszych wersjach tego komputera. Szczególnie duże niebezpieczeństwo przedwczesnego wyeksploatowania klawiatury wiąże się z wykorzystaniem mikrokomputera do gier, w których zazwyczaj będziemy używali wybiórczo tylko kilku z klawiszy. Angażując się emocjonalnie w dramaturgię gry i nieraz błyskawicznie następujące po sobie wydarzenia, wcześniej czy później zapomnimy o delikatnym naciskaniu klawiszy.

Czy miałoby to oznaczać, że z naszych zbiorów gier powinniśmy usunąć te, które są dostosowane tylko do sterowania z klawiatury?

Wręcz przeciwnie! Uważam, iż wiele z nich może pod względem swych walorów śmiało konkurować z przebojami ostatnich lat. Do niektórych z nich chętnie powracamy, chociaż nowsze gry z reguły są lepiej opracowane graficznie, dźwiękowo etc. Obserwując publikowane lawinowo przez POKE-rzystów rozmaite ułatwienia w postaci „wiecznego życia", „niezmiennej energii" lub „wstrzymania upływu czasu", chcę zaproponować wprowadzenie do programów gier niewielkich zmian, dzięki którym będziemy mogli przenieść funkcję sterowania „z klawiatury na joystick".

Czytelnikowi, pragnącemu skorzystać z moich uwag, potrzebne będą dwa „programy narzędziowe": powszechnie używany program COPY-COPY opracowany przez T. Wilczka oraz program MONS 3M i GENS 3M firmy HISOFT.

Stan poszczególnych portów odczytuje instrukcja Basica IN W. Adres W, będący liczbą z przedziału ‹0;65535›, odpowiada różnym urządzeniom wejścia. Zmiany stanu portów można badać za pomocą programu

```
10 INPUT "Adres portu = ";W
20 PRINT IN W; " ";
30 GOTO 20
```

Standardowo używa się tylko wybranych adresów, co znacznie ułatwi nam analizę gier. Oto zestawienie adresów standardowych portów:

Adres portu klawiatury		Obsługuje rząd klawiszy
dziesiętnie	szesnastkowo	
65278	FEFE	CS — V
65022	FDFE	G — A
64510	FBFE	Q — T
64386	F7FE	1 — 5
61438	EFFE	0 — 6
57342	DFFE	P — Y
49150	BFFE	ENTER — H
32766	7FFE	SPACE — B

Wprowadzając do podanego wyżej programu te adresy zauważymy, że gdy nie naciskamy żadnego z klawiszy w danym rzędzie, to wczytywany będzie bajt o wartości 191 (binarnie 1011 1111) *). Po naciśnięciu jakiegoś klawisza tego rzędu nastąpi wygaszenie odpowiadającego mu bitu (pozycja bitu wzrasta w kierunku do wnętrza rzędu). Na przykład, po wprowadzeniu instrukcji PRINT IN 65022 oraz naciśnięciu klawisza F nastąpi zgaszenie bitu 3 i odczytany bajt będzie miał wartość 187, a po równoczesnym naciśnięciu klawiszy A i G zostaną zgaszone bity 0 i 4, wskutek czego odczytana wartość bajtu będzie równa 174 **).

Podobnie, wykorzystując ten sam program, możemy sprawdzić wartość bajtu odczytywanego z portu joysticka. W przypadku używanego najczęściej joysticka typu KEMPSTON, instrukcja IN 21 odczytuje wartość 0 gdy drążek znajduje się w martwym jego położeniu lub naciśnięciu FIRE, zostanie zapalony odpowiedni bit, a więc "w prawo" — bit 0, "w lewo" — bit 1, itd. Skutkiem tego będzie odczytanie z portu joysticka wartości 1, 2, itd. ***)

Teraz poprawienie programów napisanych w języku Basic nie powinno sprawiać żadnych trudności. Odszukujemy linie z instrukcjami odczytującymi stan klawiatury, np.

```
230 IF INKEY$ = "c" THEN GOTO 270
```

Chcąc osiągnąć ten sam efekt po wychyleniu drążka joysticka "w przód", zmieniamy zapis na

```
230 IF IN 21 = 8 THEN GOTO 270
```

albo pozostawiając bez zmian instrukcję czytania klawiatury dodajemy:

```
230 IF INKEY$ = "c" OR IN 21 = 8 THEN GOTO 270
```

Niestety ciekawe gry są stosunkowo rzadko pisane w Basicu (np. MINED OUT). Znaczna ich większość ma procedury w kodzie maszynowym, w którym osiąga się dużo większą dynamikę gry i bardziej atrakcyjne rozwiązania graficzne.

Dlatego, zanim przystąpimy do omawiania konkretnego podprogramu przejścia „z klawiatury na joystick", musimy jeszcze pokrótce przedstawić sposoby odczytywania stanu klawiatury bezpośrednio z poziomu kodu maszynowego.

Najczęściej będą to rozkazy mikroprocesora Z80, odczytujące stan klawiatury poprzez standardowe porty:

Mnemonik	Kod rozkazu szesnastkowo	Adres portu w rejestrze		Wartość odczytana z portu znajduje się w rejestrze
		SB	MD	
IN A, (dd)	DB dd	A	dd	A
IN A, (C)	ED 78	B	C	A
IN H, (C)	ED 60	B	C	H
IN L, (C)	ED 68	B	C	L
IN B, (C)	ED 40	B	C	B
IN C, (C)	ED 48	B	C	C
IN D, (C)	ED 50	B	C	D
IN E, (C)	ED 58	B	C	E

Odpowiednikiem instrukcji Basica:

1) LET W = IN 65022 może być:

```
LD A, # FD          LD BC, # FDFE
                lub
IN A, (# FE)        IN A, (C)
```

(wartość odczytanego bajtu będzie w akumulatorze)

2) LET W = IN 21 może być:

```
XOR A               LD BC, # 0015
                lub
IN A, (# 15)        IN B, (C)
```

(W w akumulatorze) (W w rejestrze B)

Innym sposobem będzie wykorzystanie zmiennych systemowych, które przechowują kody ASCII, odpowiadające danemu klawiszowi. I tak, pod adresem 23556 (# 5C04) znajdziemy za pomocą funkcji PEEK kod **aktualnie naciskanego** klawisza w trybie C. W podobny sposób możemy znaleźć kod **ostatnio naciśniętego** klawisza, przechowywany w zmiennej systemowej LAST K o adresie 23560 (# 5C08). Normalnie są to kody klawiszy w trybie L, lecz równoczesne naciśnięcie któregoś z SHIFT'ów spowoduje odpowiednią zmianę wartości kodu w zmiennej LAST K.

Proszę zauważyć, że standardowo para rejestrów IY zawiera adres bazowy obszaru zmiennych systemowych, równy 23610 (# 5C3A). Dzięki temu wartość kodu z zmiennej LAST K możemy zakładać do akumulatora za pomocą rozkazu LD A, (23560), albo w trybie adresowania indeksowego LD A, (IY-50)

Kolejnym ze sposobów może być wywołanie podprogramu KEYSCAN z pamięci ROM, rozpoczynającego się od adresu 654 (# 028E). Ładuje on kod naciśniętego klawisza do pary rejestrów DE, przy czym rejestr D będzie zawierał liczbę z przedziału ‹0;39›, a rejestr E liczbę wskazującą z jakim SHIFT'em klawisz został naciśnięty. Przy równoczesnym naciśnięciu kilku klawiszy znacznik zera będzie zgaszony. Natomiast, gdy nie naciśnięto żadnego z klawiszy, w parze rejestrów DE będzie wartość #FEFE, a znacznik zera będzie zapalony. Dla zbadania wartości w rejestrach D i E oraz stanu znacznika zera można skorzystać z następującego programu

```
20 LET s=0 : For i=0 TO 18  READ a. POKE 30003+i. a
     LET s = s+a. NEXT i  IF S<> 1447 THEN PRINT "PO-
PRAW Dane"  STOP
30 PRINT  PRINT INKEY$
40 RANDOMIZE USR 30003
50 PRINT "D = ", PEEK 30001 "E = "; PEEK 30000
60 PRINT "Znacznik zera = ". PEEK 30002
70 PAUSE 100. POKE 23692.5: GOTO 30
80 DATA 205. 142,2. 237. 83. 48. 117. 40,4
90 DATA 62,0, 24,2, 61 1  50  50. 117  201
```

Program wywołuje procedurę KEY-SCAN z pamięci ROM i wpisuje wartości rejestrów D i E oraz znacznika zera do komórek 30000 — 30002 (są one różne od wartości wg kodu ASCII)

Należy na koniec zauważyć, że są to najczęściej wykorzystywane procedury odczytu stanu klawiatury w języku maszynowym. Wyszukanie w tekście programu w kodzie maszynowym odpowiedniego fragmentu „czytającego klawiaturę" i analiza jego działania jest niewątpliwie najtrudniejszym etapem pracy

Proponuję następujący algorytm postępowania.

1) Ustalić adresy startowe i długości poszczególnych bloków programu (przy pomocy programu ładującego grę albo programu COPY-COPY).

2) Zapoznać się z działaniem poszczególnych klawiszy, jakich używamy podczas gry. Zdecydować, które z działań będziemy symulować za pomocą joysticka (ruchem drążka, naciśnięciem FIRE, ewentualnie równoczesnym wykonaniem obu tych czynności).

3) Sprawdzić, czy bloki programu, realizujące daną grę, nie uruchamiają się automatycznie. Jeżeli bloki byłyby zbyt długie można je dzielić za pomocą programu COPY-COPY.

4) Wgrać blok programu, a w wolny obszar pamięci RAM program MONS 3M. Uruchomić program monitorujący i ustawić licznik główny programu na początek badanego bloku.

5) Rozkazem G wyszukać sekwencje rozkazów odczytujących stan klawiatury, wpisując na przykład.

G: DB: FE   G:ED: 78   G: 8: 5C   itp

oraz zapisać adresy tych rozkazów

6) Ustawić licznik główny programu na pierwszy ze znalezionych adresów. Jeżeli trzeba, cofnąć licznik główny o kilka komórek w celu odczytania starszego bajtu adresu portu i rejestru do jakiego będzie on załadowany, nie zmieniając przy tym desasemblacji programu. Wyświetlić stronę disasemblacji i szczegółowo przeanalizować działanie programu odczytującego stan klawiatury.

7) Przeszukać, jeżeli trzeba, pozostałe bloki programu gry w sposób podany w pp. 4 ÷ 6

8) Wybrać wolny obszar w pamięci RAM i ułożyć podprogram obsługi joysticka, nie eliminujący oryginalnej procedury sterowania z klawiatury, według następującego schematu (np. funkcję klawisza (KL) symulujemy ruchem drążka "w prawo"):

```
     ORG NN       podprogram zaczyna się od adresu
                  NN
     XOR A        wczytuje do akumulatora
     IN A,(#15)   wartość z portu joysticka
B1   BIT 0,A      sprawdza, czy drążek wychylony "w
                  prawo" (czy bit 0 równy jedynce)
     JR Z,B2      skok do kolejnego modułu, gdy wa-
                  runek nie spełniony (bit 0 równy
                  zero)
     LD A,d       ładuje do akumulatora wartość kodu
                  d, jaka byłaby odczytana według pro-
                  cedury w programie głównym po na-
                  ciśnięciu klawisza KL
     RET          powrót do programu głównego i wy-
                  konania podprogramu obsługi klawi-
                  sza KL
B2   BIT 1,A      czy drążek wychylony "w lewo" —
                  początek modułu symulacji następ-
                  nego klawisza



C:                linie programu głównego, odczytują-
                  ce stan klawiatury, zastąpione w nim
                  przez CALL NN
     RET          powrót do programu głównego
```

Przy układaniu tego podprogramu wygodne będzie użycie programu GENS 3M, za pomocą którego wygenerowany kod wynikowy wygrywamy na taśmę

9) Zastąpić w analizowanym bloku wszystkie rozkazy odczytujące stan danego klawisza rozkazem wywołania podprogramu obsługi joysticka CALL NN (stan tego samego klawisza może być odczytywany w kilku różnych miejscach programu). Poprawiony blok nagrać na taśmę.

10) Wmontować opracowany podprogram obsługi joysticka do poprawionego programu gry za pomocą programu COPY-COPY albo z poziomu języka Basic za pomocą instrukcji POKE (pamiętając o tym, aby nie kolidował on z obszarem programu głównego, a także aby został wczytany do pamięci RAM przed uruchomieniem się gry).

11) Sprawdzić poprawność działania poprawionego programu

Za miesiąc podam sposoby przystosowania kilku popularnych gier do sterowania joystickiem.

Adam Gawłowski

---

*) albo 31 (binarnie 0001 1111) w przypadku mikrokomputera TIMEX 2048

**) w przypadku mikrokomputera TIMEX 2048 będą to wartości 23 i 14 odpowiednio

***) dla joysticka innego typu adresy portów i wartości odczytanego bajtu będą inne, co należy ustalić przed przystąpieniem do przeróbek

NIE TYLKO DLA NERWOWYCH GRACZY

Teoretycznie mogłoby się wydawać, iż drążek jest elementem wytrzymałym na wszystko i tak prostym, że w żaden sposób nie może zaszkodzić komputerowi. Jak mylny jest to pogląd wiedzą chyba wszyscy użytkownicy, którzy zmuszeni byli do wymiany jednego z układów scalonych komputera i to bynajmniej wcale nie najtańszego...

We wszystkich komputerach Commodore drążki sterowe są obsługiwane przez specjalizowane układy interfejsowe: w C-64 i 128 są to układy 6526 CIA (Complex Intreface Adapter), w VIC-20 układ VIA (6522), w rodzinie C-16/116 i PLUS/4 (układ 7360 (TED) wraz z układem 6529B. Zupełnie nie wiadomo dlaczego przyjęło się, że nie wolno odłączać podczas pracy komputera jakichkolwiek urządzeń peryferyjnych z wyjątkiem drążków. W efekcie delikatne struktury układów nie wytrzymują iskrzeń obecnych przy wkładaniu wtyku drążka do innego portu czy palą się pod wpływem ładunków elektrostatycznych.

Najczęstszym objawem uszkodzenia takiego układu jest nagły brak kontaktu np. przycisku FIRE czy niemożność przesunięcia sterowanego obiektu w jakąś stronę. Jeżeli więc po przetestowaniu drążka stwierdzimy, że jest on całkowicie sprawny oznacza to, że spaleniu uległa sekcja układu scalonego i trzeba go będzie wymienić. Niech się nikomu nie wydaje, że jest to wyjątkowo proste: sam układ kosztuje za granicą ok. 15 dolarów i jest raczej trudnodostępny (własna produkcja Commodore). W Polsce sam układ CIA kosztuje w chwili gdy to piszę ok. 50 000 zł (bez naprawy) i konia z rzędem temu, kto go dostanie przy pierwszym podejściu (nawet na giełdzie).

Jak z tego widać warto dbać również o tak prymitywny element jak drążek. Jeżeli będziesz stosował się do poniżej podanych zasad nie powinieneś mieć z drążkiem żadnych problemów. Warto o tym pamiętać ponieważ przydaje się on nie tylko do gier ale wykorzystują go też znacznie ciekawsze programy takie jak GEOS, NEWSROOM, CERTIFICATE MAKER, PRINT MASTER PLUS i wiele innych.

1. **NIGDY nie przełączaj (przyłączaj) drążka w czasie gdy komputer jest włączony.** Najprostszym rozwiązaniem jest zakup drugiego drążka i przyłączenie obu na stałe do komputera; ich przełączenie nie będzie wtedy konieczne.

2. Przelewanie emocji grającego na styki joysticka skraca jego życie o co najmniej połowę (styki są oczywiście niedostępne). **Jeżeli nie jesteś w stanie się opanować, spróbuj nabyć drążek pracujący w oparciu o mikrowyłączniki** (są już takie drążki rodzimej produkcji w cenie ok. 5000 — 6000 zł.

3. **Staraj się zawsze tak ustawiać komputer, aby uniemożliwić przypadkowe dotknięcie ręką do portów obu drążków** (dotyczy to także wszystkich innych gniazd).

4. **Nie używaj drążków uszkodzonych czy niepewnych, nie szarp też za przewód połączeniowy.** Jeśli Twój drążek ma gumowe przysawki i chcesz go przesunąć, unoś go zawsze za obudowę, a nigdy za rączkę.

5. Przy przenoszeniu joysticka **nie owijaj przewodu dookoła rączki**, gdyż może to spowodować wewnętrzne pęknięcie jednej z jego żył.

6. Jeżeli Twój drążek jest wyposażony w przycisk AUTOFIRE, to **sprawdzaj każdorazowo przed włączeniem komputera, czy jest on wyłączony.** Ruchy drążkiem przyłączonym do portu są odbierane przez układ CIA i jak normalnie wciśnięte klawisza, tak więc poruszenie rączką może spowodować ukazanie się na ekranie dziwnych znaków, liter, cyfr itp.

Klaudiusz Dybowski

PRZEDSTAWIAMY WARSAW BASIC(3)

ORGANIZACJA PAMIĘCI

Za konfigurację pamięci komputera C-64, wykorzystywaną przez interpreter CBM V2.0, odpowiedzialne są zmienne systemowe, których wartości przechowywane są od $0000 do $0333 i od $d000 do $dfff.

O tym jaka część pamięci przeznaczona jest na tekst programu, zmienne, tablice i wartości wyrażeń tekstowych decydują zmienne systemowe: $2b/$2c — początek programu (TXTTAB). $2d/$2e — początek zmiennych (VARTAB). $2l/$30 — początek tablic (ARYTAB). $31/$32 — koniec tablic (STREND). — $33/$34 — koniec wartości wyrażeń tekstowych (FRETOP) i $37/$38 — początek wartości wyrażeń tekstowych i koniec dostępnej pamięci (MEMTOP).

Po inicjacji interpretera CBM V2.0, dla programu dostępna jest pamięć rozpoczynająca się od adresu $0800, pod kórym jest wstawiona wartość $00. Zmienna TXTTAB wskazuje na adres $0801, który jest właściwym początkiem programu. Linie programu są zapisywane od tego adresu w formacie: wskaźnik do początku następnej linii (2 bajty), numer linii (2 bajty), treść linii (nie więcej niż 79 znaków) i koniec linii (1 bajt $00). Interpreter rozpoznaje koniec linii programu po napotkaniu zerowego wskaźnika ($00 $00) do początku następnej linii. W standardowym interpreterze następny adres po tym wskaźniku przechowuje zmienna VARTAB. Koniec zmiennych pokrywa się z początkiem tablic (ARYTAB). Deklaracja lub zakładanie nowych zmiennych powoduje przesuwanie się w kierunku wyższych adresów końca tablic (STREND). W chwili uruchomienia programu MEMTOP pokrywa się z FRETOP. W miarę przybywania wyników działań na tekstach zawartość FRETOP przesuwa się w kierunku niższych adresów.

Jak wiemy, jeżeli założenie nowej zmiennej spowodowałoby przekroczenie FRETOP przez STREND, to sygnalizowany jest błąd OUT OF MEMORY (brak pamięci). Nie wszyscy jednak wiedzą, że obniżenie FRETOP poniżej STREND (próba zapisania nowej wartości wyrażenia tekstowego) nie jest sygnalizowane. W tym przypadku interpreter przesuwa FRETOP do pozycji wyjściowej (traci się dostęp do wcześniej wypracowanych tekstów).

Interpreter CBM V2.0 toleruje również inne usytuowanie względem siebie swoich danych (programu i zmiennych). Początek tekstu programu można umieszczać w dowolnym, dostępnym miejscu w pamięci. Koniec programu nie musi pokrywać się z początkiem zmiennych (VARTAB). Co więcej, tekst programu może być umieszczony za zmiennymi nawet powyżej adresu wskazywanego przez MEMTOP.

W Warsaw BASIC-u te różne konfiguracje są wykorzystywane np. do organizowania bloku zmiennych lokalnych, wspólnego bloku zmiennych, RAM dysku, parametrów formalnych, przekazywania wartości parametrów przez wartość lub nazwę, wywoływania procedur, ich zagnieżdżania, automatycznego zestawiania procedur w pakiety, nakładkowania itp. Dzieje się to wszystko w sposób niewidoczny dla użytkownika.

Przy pracy z procedurami można wyróżnić kilka poziomów pamięci. Na poziomie 0 jest program główny, wczytywany dynamicznie w trybie bezpośrednim i jego zmienne, zakładane w trybie programowym. Procedury wywoływane przez program główny, wczytywane dynamicznie w trybie programowym i ich zmienne (lokalne) zajmują poziom 1. Początek nowego poziomu znajduje się bezpośrednio za końcem tablic poziomu poprzedniego. Aby w każdej chwili można było rozpoznać aktualną konfigurację pamięci na danym poziomie, w Warsaw BASIC-u wprowadzono instrukcję MEM(ory) (patrz BAJTEK 7/87). Na użytek czytelników naszego cyklu podajemy uproszczoną wersję tej instrukcji, wywoływaną przez £m. Program 2 umożliwia, po dołączeniu do programu z poprzedniego odcinka i wykonaniu tej instrukcji, odczytanie liczby wolnych bajtów na poziomie 0

Program organizujący inne poziomy przedstawimy w jednym z następnych odcinków. Wtedy instrukcja £m będzie również pomocna w czasie „ręcznej" wędrówki po różnych poziomach.

Krzysztof Gajewski
Bogusław Radziszewski

```
                ORG   $C475
TXTAB           -     $002B
                LDY   #$C4
                LDA   #$A3
                JSR   HAB1E
                LDX   #$06
                LDY   #$08
                LDA   TXTAB.Y
                SEC
                SBC   $2B.X
                STA   $FE
                LDA   TXTAB+1.Y
                SBC   $2C.X
                LDX   $FE
                STY   $D3
                JSR   HBDCD
                LDA   #$CB
                LDY   #$C4
                JSR   HAB1E
                LDA   HC4CA
                ORA   #$30
                JMP   CHROUT
                HEX   93A0A0A0A0A0A0A012
                HEX   2AA0C94E5445525052 45
                HEX   544552A0302E30A02AA0
                HEX   0D0D574F4C4E453A0000
                HEX   0D504F5A494F4D3AA000
HAB1E           -     $AB1E
HBDCD           -     $BDCD
HC4CA           -     $C4CA
CHROUT          -     $FFD2
```

```
200 print [illegible]
205 x=50293:n=95:c=0
210 fori=0ton:reada:pokex+i,a:c=c+a:next
215 ifc=10526thenprint" [illegible] :goto225
220 print" [illegible] :end
225 poke50265,7*16+4:poke50266,12*16+4
230 data160,196,169,163,32,30.171,162,6,160,8,185
235 data43,0,56,245,43,133,254,185,44,0,245,44
240 data166,254,132,211,32,205,189,169,203,160,196,32
245 data30,171,173,202,196,9,48,76,210,255,147,160
250 data160,160,160,160,160,160,160,18,42,160,201,78
255 data84,69,82,80,82,69,84,69,82,160,48,46
260 data48,160,42,160,13,13,87,79,76,78,69,58
265 data0,0,13,80,79,90,73,79,77,58,160,0
```

ELEKTRONIKA JOYSTICKA

Wspominałem już, że przekładanie drążka z portu do portu nie jest ani bezpieczne dla komputera ani też nie przyczynia się do przedłużenia żywotności układów scalonych obsługujących drążki. Jako dowód przedstawiam czytelnikom schematy obrazujące sposób przyłączenia obu portów do poszczególnych układów scalonych w C-128, C-64 i C-16/116/PLUS/4.

Schematy te mogą służyć także do jeszcze innego celu. Jak wiadomo na rynku są powszechnie dostępne drążki z wtykiem dziewięciobolcowym czyli do C-64, C-128, Atari, nie ma natomiast drążków do rodziny komputerów C-16. Na rysunku 4 przedstawiono wyprowadzenia obu portów w C-16/116/PLUS/4 oraz C-64/C-128. Kontakt powoduje zwarcie jednej z linii JOYA0-JOYA3 lub JOYB0-JOYB3 z masą układu dostępną na nóżce 8 w C-64/128 WB z nóżką 8 (w C-16, 116, PLUS/4). Jeżeli więc ktoś ma smykałkę w ręku wykona we własnym zakresie odpowiednią wtyczkę (lub nabędzie takową na giełdzie) i będzie w stanie przerobić zwykły drążek w drążek do C-16. Poszczególne linie oznaczają:

Nóżka 1 gniazda — JOYA0 (JOYB0) — góra dla C-64-128, dół, dla rodziny C-16

Nóżka 2 gniazda — JOYA1 (JOYB1) — dół, dla C-64/R8, góra dla rodziny C-16

Nóżka 3 gniazda — JOYA2 (JOYB2) — lewo,

Nóżka 4 gniazda — JOYA3 (JOYB3) — prawo,

Nóżka 5 gniazda — nie dotyczy drążka w C-64/128,

Nóżka 5 gniazda — +5V w C-16/116/PLUS/4,

Nóżka 6 gniazda — BTNA (BTNB) — FIRE,

Nóżka 7 gniazda — masa w C-16/116/PLUS/4,

Nóżka 7 gniazda — +5V w C-64/128, korzysta też z tego AUTOFIRE,

Nóżka 8 gniazda — masa w C-64/128,

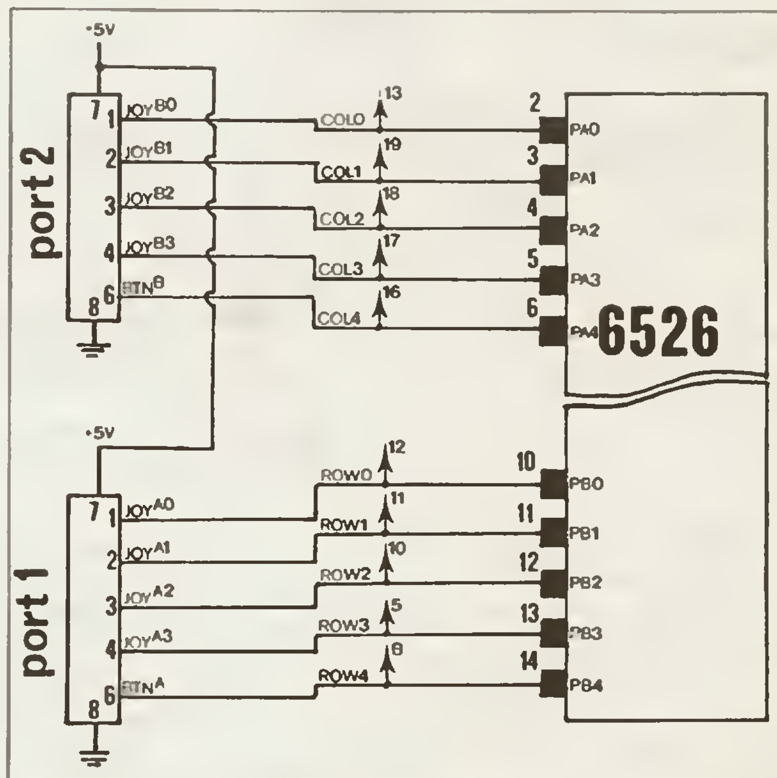
Nóżka 8 gniazda — linia sygnałowa C-16/116/PLUS/4

Nóżka 9 gniazda — nie dotyczy drążka w C-64/128.

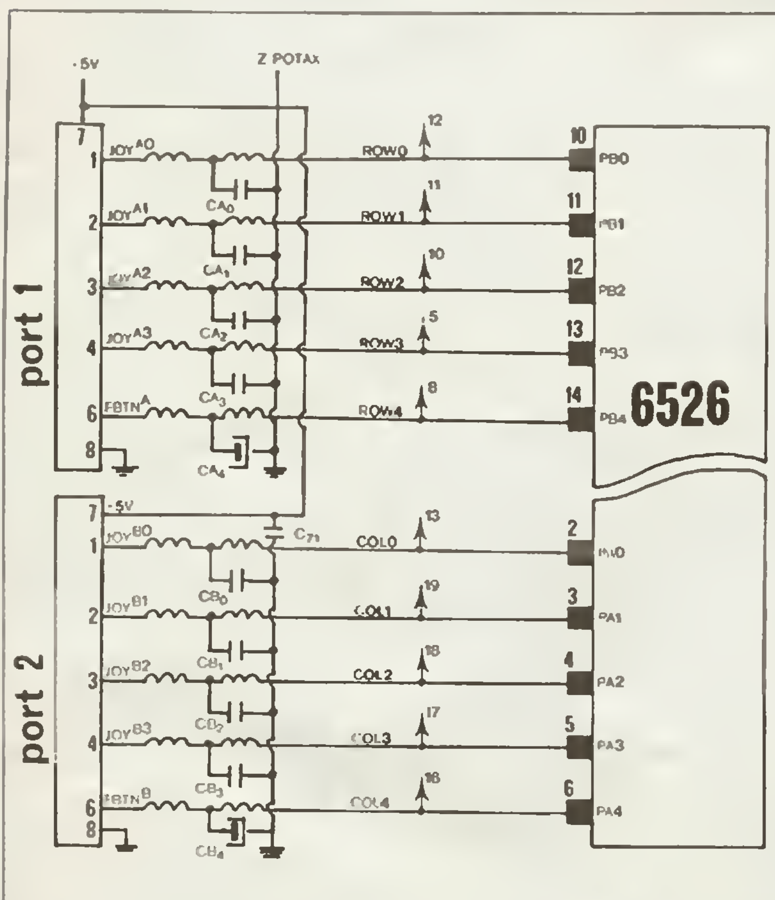
Jeżeli wśród czytelników są tacy, którzy początkują dopiero jako elektronicy-amatorzy lub woleliby prostsze rozwiązanie to informuję, że czasami można dostać na giełdzie odpowiednie wtyki przejściowe umożliwiające przyłączenie drążków standardowych do rodziny komputerów C-16. Nowicjuszom polecałbym asystę fachowców podczas zmiany wtyków drążka.

Ze schematów można także odczytać, iż najbardziej podatnym na uszkodzenia układu obsługującego drążek jest Commodore 64 — gniazdo tego portu jest połączone BEZPOŚREDNIO z układem bez żadnych buforów jak ma to miejsce w C-128 czy rodzinie C-16. Zwracam więc jeszcze raz uwagę użytkowników, aby nie przełączać drążków gdy komputer jest włączony.

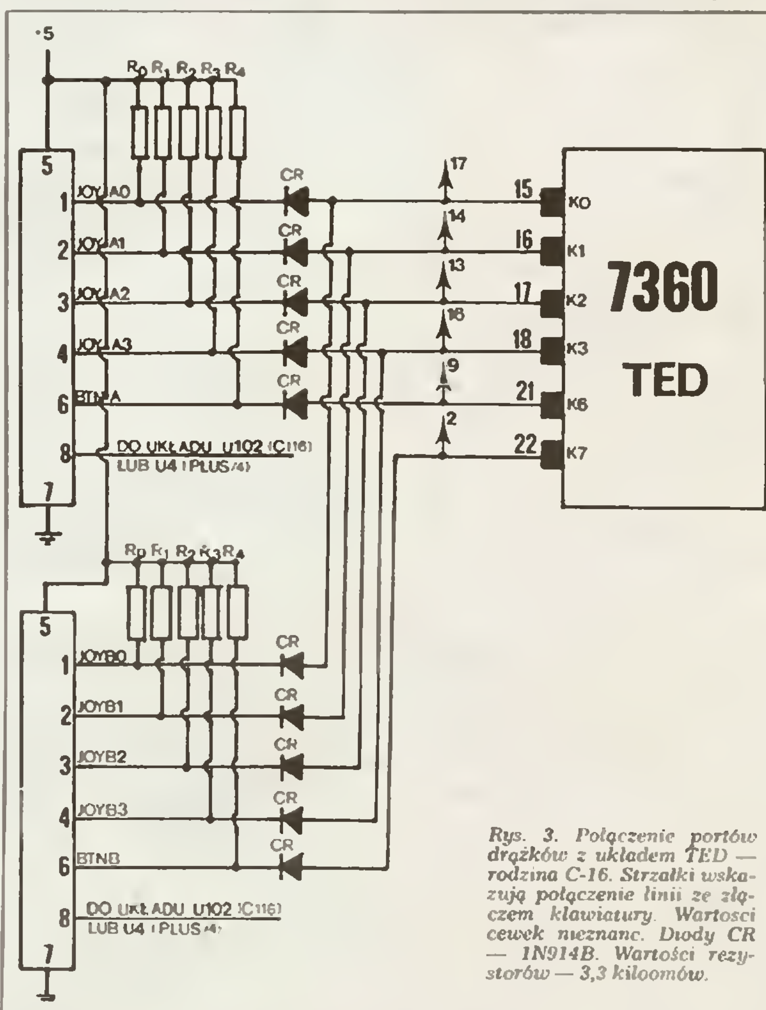
Klaudiusz Dybowski



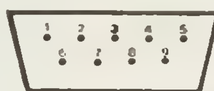
Rys. 2. Połączenie portów drążków z układem CIA — Commodore 128. Strzałki wskazują połączenie linii ze złączem klawiatury. Wartości cewek nieznane. Wartości kondensatorów CA0-CA3 i CB0-CB3 = 100 pikofaradów, C71 — 0,1 mikrofarada.



Rys. 1. Połączenie portów drążków z układem CIA — Commodore 64. Strzałki wskazują połączenie linii ze złączem klawiatury.



Rys. 3. Połączenie portów drążków z układem TED — rodzina C-16. Strzałki wskazują połączenie linii ze złączem klawiatury. Wartości cewek nieznane. Diody CR — 1N914B. Wartości rezystorów — 3,3 kiloomów.





Rys. 4 Gniazda portów drążków sterowych a) C-64/128 b) C-16/116/PLUS/4

JOYTEST

Zabawa w gry komputerowe bywa czasami bardzo kosztowna wystarczy tylko jedno przełączenie drążka z portu do portu w czasie gdy komputer jest włączony... Znacznie częściej jednak usterce ulegają styki drążka, pękają membranki czy blaszki sprężynujące.

Jak sprawdzić co uległo uszkodzeniu — komputer czy drążek? Bardzo prosto. Wystarczy w tym celu wpisać i uruchomić jeden z podanych poniżej programów, w zależności od posiadanego komputera. Można oczywiście zrobić to także wczytując pierwszą lepszą grę, pewniejsze są jednak podane tu programy. Oprócz samej animacji ruchu drążkiem na ekranie użytkownik otrzymuje także informację o sygnale wejściowym generowanym przez drążek. Dzięki temu możemy np. określić, czy styki są czyste i kontaktują za każdym razem, czy AUTOFIRE generuje regularny sygnał itp.

Przed uruchomieniem programu warto go będzie zapisać na taśmie czy dysku, należy też pamiętać o tym, aby drążek był podłączony PRZED włączeniem komputera. Przeprowadzenie testu jest możliwe zarówno z portu 1 jak i 2, bez różnicy.

Klaudiusz Dybowski

```
52   100 REM ### JOYTEST 16/116/PLUS/4  ###
38   105 :
AF   110 REM ###      K. DYBOWSKI       ###
F8   115 :
78   120 :
B1   125 SCNCLR:GRAPHIC CLR:CHAR 1,9,12,"PORT
          1 CZY 2 (1/2) : ":INPUT P
45   130 IF P<1 OR P>2 GOTO 125
BC   135 P$="FUNKCJA JOY("+RIGHT$(STR$(P),1)+
         ") = ":E$=CHR$(209)
23   140 COLOR 1,1:COLOR 0,2:GRAPHIC 1,1
3B   145 :
D9   150 FOR I=1 TO 4:READ K,L,M,N:BOX 1,K,L,
         M,N:NEXT
DC   155 FOR I=0 TO 135 STEP 45:BOX 1,128,95,
         198,105,I,1:NEXT
B6   160 BOX 1,230,50,240,170,,1:BOX 1,113,16
         5,240,175,,1:CIRCLE 1,204,52,8,8
BC   165 :
FE   170 A=JOY(P):W=0:CHAR 1,X,Y," ":CHAR 1,2
         5,6," ":X=20:Y=12
55   175 IF A=0 THEN CHAR1,20,12,E$:GOTO 230
71   180 IF A=>128 THEN W=128:A=(A AND 127):C
         HAR 1,25,6,E$
47   185 IF A=1 THEN X=20:Y=7
0A   190 IF A=2 THEN X=24:Y=8
A6   195 IF A=3 THEN X=25:Y=12
BB   200 IF A=4 THEN X=24:Y=16
84   205 IF A=5 THEN X=20:Y=17
49   210 IF A=6 THEN X=16:Y=16
C5   215 IF A=7 THEN X=15:Y=12
ED   220 IF A=8 THEN X=16:Y=8
CF   225 :
95   230 A$=STR$(A+W):IFLEN(A$)=4THENA$=RIGHT
         $(A$,3):ELSE:A$="00"+RIGHT$(A$,1)
ED   235 CHAR 1,10,1,P$+A$,1:CHAR 1,X,Y,E$
73   240 GOTO 170
51   245 :
6E   250 DATA 097,038,226,162,100,040,223,160
29   255 DATA 103,042,220,158,094,036,229,164
```

```
8F   100 REM   ###   JOYTEST 128   ###
38   105 :
AF   110 REM   ###   K. DYBOWSKI   ###
F8   115 :
78   120 :
B1   125 SCNCLR:GRAPHIC CLR:CHAR 1,9,12,"PORT 1
          CZY 2 (1/2) : ":INPUT P
45   130 IF P<1 OR P>2 GOTO 125
40   135 P$="FUNKCJA JOY("+RIGHT$(STR$(P),1)+")
          = ":GOSUB 260
23   140 COLOR 1,1:COLOR 0,2:GRAPHIC 1,1
04   145 X1=160:Y1=130:X2=197:Y2=90:X3=125:Y3=1
         68
BA   150 :
F7   155 FOR X=1 TO 4:READ K,L,M,N:BOX 1,K,L,M,
         N:NEXT:BOX 1,150,145,170,153,1
7C   160 FOR X=0 TO 135 STEP 45:BOX 1,125,95,19
         5,105,X,1:NEXT
F7   165 MOVSPR 1,X1,Y1:MOVSPR 2,X1,Y1
EA   170 BOX 1,227,50,237,170,,1:BOX 1,107,165,
         237,175,,1
7D   175 :
C9   180 A=JOY(P):W=0:SPRITE 1,1,8,0,1,1:SPRITE
          2,1,8,0,1,1
13   185 IF A=0 OR A=128 THEN MOVSPR 1,X1,Y1:MO
         VSPR 2,X1,Y1
F2   190 IF A=>128 THEN W=128:SPRITE 2,1,3,0,1,
         1:A=(A AND 127)
07   195 IF A=1 THEN MOVSPR 1,X1,Y2:MOVSPR 2,X1
         ,Y2:GOTO 240
95   200 IF A=2 THEN MOVSPR 1,X2,Y2:MOVSPR 2,X2
         ,Y2:GOTO 240
D7   205 IF A=3 THEN MOVSPR 1,X2,Y1:MOVSPR 2,X2
         ,Y1:GOTO 240
DF   210 IF A=4 THEN MOVSPR 1,X2,Y3:MOVSPR 2,X2
         ,Y3:GOTO 240
B9   215 IF A=5 THEN MOVSPR 1,X1,Y3:MOVSPR 2,X1
         ,Y3:GOTO 240
81   220 IF A=6 THEN MOVSPR 1,X3,Y3:MOVSPR 2,X3
         ,Y3:GOTO 240
BE   225 IF A=7 THEN MOVSPR 1,X3,Y1:MOVSPR 2,X3
         ,Y1:GOTO 240
5D   230 IF A=8 THEN MOVSPR 1,X3,Y2:MOVSPR 2,X3
         ,Y2
90   235 :
D6   240 A$=STR$(A+W):IF LEN(A$)=4 THEN A$=RIGH
         T$(A$,3):ELSE:A$="00"+RIGHT$(A$,1)
9C   245 CHAR 1,10,1,P$+A$,1
04   250 GOTO 180
12   255 :
0F   260 FOR J=3584 TO 3711:READ Q$:POKE J,DEC(
         Q$):NEXT:RETURN
D2   265 :
99   270 DATA 00,7E,00,01,FF,80,07,FF,E0,0F,FF,
         F0,1F,FF,F8,3F
EA   275 DATA FF,FC,3F,FF,FC,7E,00,7E,7C,00,3E,
         FC,00,3F,FC,00
53   280 DATA 3F,FC,00,3F,7C,00,3E,7E,00,7E,3F,
         FF,FC,3F,FF,FC
3A   285 DATA 1F,FF,F8,0F,FF,F0,07,FF,E0,01,FF,
         80,00,7E,00,00
1B   290 DATA 00,7E,00,01,FF,80,07,FF,E0,0F,FF,
         F0,1F,FF,F8,3F
DD   295 DATA FF,FC,3F,FF,FC,7F,FF,FE,7F,FF,FE,
         FF,FF,FF,FF,FF
5A   300 DATA FF,FF,FF,FF,7F,FF,FE,7F,FF,FE,3F,
         FF,FC,3F,FF,FC
6A   305 DATA 1F,FF,F8,0F,FF,F0,07,FF,E0,01,FF,
         80,00,7E,00,00
04   310 :
3B   315 DATA 097,038,223,162,100,040,220,160
6A   320 DATA 103,042,217,158,094,036,226,164
```

```
1A   100 REM   ###   JOYTEST 64   ###
38   105 :
AF   110 REM   ###   K.DYBOWSKI   ###
F8   115 :
78   120 :
95   125 S$=CHR$(147):PRINTS$:PRINT:INPUT "PORT
          1 CZY 2 (1/2) : ";P:C$=CHR$(34)
09   130 IF P=1 THEN P=56321:GOTO 145
B8   135 IF P=2 THEN P=56320:GOTO 145
4D   140 GOTO 125
AF   145 P$="WARTOSC "+C$+"NOT PEEK("+RIGHT$(ST
         R$(P),5)+") AND 31"+C$+" = "
CD   150 PRINT S$:GOSUB 265
C3   155 X1=160:Y1=130:X2=197:Y2=90:X3=125:Y3=1
         68:V=53248:S1=V+1:S2=V+2:S3=V+3
2E   160 POKEV+32,6:POKEV+33,1:POKE 646,0
BC   165 :
6D   170 POKEV+39,7:POKEV+40,7:POKEV+23,0:POKEV
         +29,3:POKEV+23,3:POKEV+21,3
7D   175 :
D4   180 A= NOT PEEK(P) AND 31:W=0:POKEV+40,7
32   185 IF A=0 OR A=16 THEN POKEV,X1:POKES1,Y1
         :POKES2,X1:POKES3,Y1
AA   190 IF A=>16 THEN W=16:POKEV+40,2:A=(A AND
          15)
88   195 IF A=1 THEN POKEV,X1:POKES1,Y2:POKES2,
         X1:POKES3,Y2:GOTO 240
96   200 IF A=9 THEN POKEV,X2:POKES1,Y2:POKES2,
         X2:POKES3,Y2:GOTO 240
5A   205 IF A=8 THEN POKEV,X2:POKES1,Y1:POKES2,
         X2:POKES3,Y1:GOTO 240
71   210 IF A=10 THEN POKEV,X2:POKES1,Y3:POKES2
         ,X2:POKES3,Y3:GOTO 240
56   215 IF A=2 THEN POKEV,X1:POKES1,Y3:POKES2,
         X1:POKES3,Y3:GOTO 240
4F   220 IF A=6 THEN POKEV,X3:POKES1,Y3:POKES2,
         X3:POKES3,Y3:GOTO 240
95   225 IF A=4 THEN POKEV,X3:POKES1,Y1:POKES2,
         X3:POKES3,Y1:GOTO 240
D2   230 IF A=5 THEN POKEV,X3:POKES1,Y2:POKES2,
         X3:POKES3,Y2
90   235 :
E5   240 Z=A+W:IF LEN(STR$(Z))=2 THEN Z$="0"+RI
         GHT$(STR$(Z),1):GOTO 250
46   245 Z$=RIGHT$(STR$(Z),2)
2E   250 PRINTTAB(1);P$Z$CHR$(145)CHR$(157)CHR$
         (32)
C6   255 GOTO 180
91   260 :
F5   265 FOR J=832 TO 959:READ Q:POKE J,Q:NEXT:
         POKE 2040,13:POKE 2041,14
72   270 FOR I=1 TO 14:T=T+40:FOR X=55510+T TO
         55521+T:POKEX,0:POKEX-54272,160:NEXT
98   275 NEXT:RETURN
13   280 :
18   285 DATA 000,126,000,001,255,128,007,255,2
         24,015,255,240,031,255,248,063
67   290 DATA 255,252,063,255,252,126,000,126,1
         24,000,062,252,000,063,252,000
BF   295 DATA 063,252,000,063,124,000,062,126,0
         00,126,063,255,252,063,255,252
B2   300 DATA 031,255,248,015,255,240,007,255,2
         24,001,255,128,000,126,000,000
48   305 DATA 000,126,000,001,255,128,007,255,2
         24,015,255,240,031,255,248,063
50   310 DATA 255,252,063,255,252,127,255,254,1
         27,255,254,255,255,255,255,255
A0   315 DATA 255,255,255,255,127,255,254,127,2
         55,254,063,255,252,063,255,252
2D   320 DATA 031,255,248,015,255,240,007,255,2
         24,001,255,128,000,126,000,000
```

KOREKTA KYAN PASCALA

Jednym z lepszych języków dla komputera ATARI — 800XL jest Pascal opracowany przez firmę KYAN.

Dysponuję wersją 1.1 tego języka, która niestety obarczona jest pewną wadą utrudniającą pracę.

Wadę tę demonstruje program 1. Proszę wprowadzić go do komputera, skompilować i uruchomić — będzie on wczytywał dwie liczby rzeczywiste i drukował ich sumę, różnicę, iloczyn i iloraz. Następnie proszę wprowadzić dane:

A = 10

B = 0

Niestety — na ekranie nie pojawi się oczekiwany komunikat „ARITHMETIC OVERFLOW" (spowodowany dzieleniem przez zero), zamiast tego z reguły ekran gaśnie i system zawiesza się.

```
PROGRAM PRZYKLAD1;
 VAR A,B,C:REAL;
  D:INTEGER;
BEGIN
  D:=1;
  REPEAT
    WRITE('A=');READLN(A);
    WRITE('B=');READLN(B);
    C:=A+B;WRITELN('A+B=',C);
    C:=A-B;WRITELN('A-B=',C);
    C:=A*B;WRITELN('A*B=',C);
    C:=A/B;WRITELN('A/B=',C);
  UNTIL D=0
END.
```

Przyczyna tego niemiłego faktu jest dość prosta: Obliczenia numeryczne wykorzystują tak zwany dziesiętny tryb pracy procesora. Gdy wystąpi błąd obliczeń, zostaje wywołana procedura obsługi błędów (zawarta w zbiorze LIB), która między innymi wykorzystuje procedurę systemu operacyjnego, umieszczającą znaki na ekranie monitora. Procedura ta działa błędnie, gdy wykonywana jest przez procesor pracujący w trybie dziesiętnym i zwykle doprowadza do zawieszenia systemu. Przyczyną błędu jest więc brak instrukcji CLD, przywracającej normalny tryb pracy procesora, przed wywołaniem procedury drukującej tekst na ekranie.

Powyższa diagnoza nie byłaby wiele warta gdybym nie podał od razu odpowiedniej „kuracji". Oto

```
PROGRAM PRZYKLAD2;
 VAR A,B,C:REAL;
  D:INTEGER;
BEGIN
#A
 LDA #0
 STA 39020
 LDA #216
 STA 39021
 LDA #<39021
 STA 40398
 LDA #>39021
 STA 40399
#
  D:=1;
  REPEAT
    WRITE('A=');READLN(A);
    WRITE('B=');READLN(B);
    C:=A+B;WRITELN('A+B=',C);
    C:=A-B;WRITELN('A-B=',C);
    C:=A*B;WRITELN('A*B=',C);
    C:=A/B;WRITELN('A/B=',C);
  UNTIL D=0
END.
```

i ona. Na początku procedury obsługi błędów arytmetycznych należy dopisać instrukcję CLD. Adres startu jej procedury ulegnie przez to zmianie, ale na szczęście jest ona wywoływana zawsze pośrednio, przez tablicę skoków — wystarczy więc zmienić adres w tej tablicy. To wszystko realizuje procedura w kodzie maszynowym zawarta w programie 2.

Niestety, podobnie jak większość lekarstw, także i to ma działanie uboczne: dodatkowa instrukcja została dopisana kosztem uszkodzenia komunikatu błędu „RANGE ERROR", będzie on teraz drukowany jako „RANGE ERRO". Wydaje mi się to jednak niezbyt wysoką ceną.

Podaną tu procedurę można wykorzystywać we własnych programach. Powinna być ona pierwszym wykonywanym fragmentem programu. Jeśli komuś takie rozwiązanie nie odpowiada, to podaję inne, pozwalające trwale poprawić zbiór „LIB" Pascala będący przyczyną tych kłopotów.

Program 3, napisany w Turbo Basicu XL należy uruchomić umieszczając w stacji dysków kopię dyskietki z Pascalem.

Przypominam, że podane tu poprawki dotyczą jedynie wersji 1.1. KYAN Pascala

Życzę przyjemnej pracy z Pascalem już nie bojącym się błędów!

Wojciech Zabołotny

```
10 REM PROGRAM 3
20 REM Poprawia zbior "D:LIB"
30 REM zapisany na dyskietce.
40 REM Radze sprobowac najpierw na
50 REM kopii dyskietki z Pascalem!
60 DIM A$(9716):A$(9716)=" "
70 OPEN #1,4,0,"D:LIB"
80 BGET #1,ADR(A$),9716
90 CLOSE #1
100 A$(1139,1139)=CHR$(0)
110 A$(1140,1140)=CHR$(216):REM CLD
120 A$(2517,2517)=CHR$(39021 MOD 256)
130 A$(2518,2518)=CHR$(39021 DIV 256)
140 OPEN #1,8,0,"D:LIB"
150 BPUT #1,ADR(A$),9/16
160 CLOSE #1:END
```

ZAMIANA LICZB

Z przykrością muszę stwierdzić, że nie znalazłam dotąd w „Bajtku" artykułu (programu) napisanego przez DZIEWCZYNĘ. Chciałabym zmienić ten stan rzeczy proponując program, którego zadaniem jest zamiana liczb z systemu dziesiętnego na liczby przedstawione w systemie dwójkowym i szesnastkowym (wbudowany BASIC nie posiada takiej funkcji).

w innym systemie, a polega on na dzieleniu zadanej wartości przez podstawę określonego systemu. Otrzymane z dzielenia reszty będą kolejnymi cyframi reprezentacji zadanej wartości w tym systemie.

Brzmi to trochę dziwnie i niezrozumiale. Przedstawmy przykład, który pozwoli lepiej nam zrozumieć ten sposób konwersji.

Chcemy przedstawić liczbę 11 w systemie dwójkowym. Dzielimy zatem 11 przez 2 (podstawę systemu dwójkowego).

11:2=5 r.1
5:2=2 r.1
2:2=1 r.0
1:2=0 r.1

Otrzymane reszty (r.) piszemy od pozycji najmniej znaczącej (od końca) i w ten sposób otrzymujemy liczbę 11 przedstawioną w systemie dwójkowym.

11 – –> 1011

Analogicznie rzecz się przedstawia przy zamianie liczby systemu dziesiętnego na liczbę w systemie szesnastkowym z tą tylko różnicą, że podstawa dzielenia jest liczbą 16, a dziesiętne wartości równe 10,11,12,13,14,15 są przedstawione literami A,B,C,D,E,F.

Dla lepszego zrozumienia przykład:
Zamieniamy liczbę 8360 (w systemie dziesiętnym) na liczbę przedstawioną w systemie szesnatkowym.

8360:16=522 r.8
522:16=32 r.10 — A
32:16=2 r.0
2:16=0 r.2

Przepisanie wartości reszt od pozycji najmniej znaczącej da nam liczbę 8360 przedstawioną w systemie szesnastkowym (należy pamiętac, że wartość 10 odpowiada litera A). Ostatecznie otrzymujemy:

[illegible]360 – – [illegible]A8

```
JJ   1 REM *** LICZBY ***
IE   5 DIM A(16),A$(1),D(4),Z$(1)
NO   10 TRAP 190
VM   15 ? :? "LICZBA"
OH   20 ? "w systemie dziesietnym ";
JP   25 INPUT B
ZW   30 IF B>65535 THEN 180
YO   35 Z=B
JW   40 FOR I=1 TO 16
WU   45 A=B/2
VJ   50 C=INT(A)
GF   55 IF A<>C THEN T=1
LW   60 IF A=C THEN T=0
RD   65 A(17-I)=T
UW   70 B=C
JF   75 NEXT I
UI   80 ? "w systemie dwojkowym",
KK   85 FOR I=1 TO 16
GO   90 ? A(I);
JH   95 NEXT I
CE   100 ?
NM   105 FOR I=1 TO 4
IF   110 C=INT(Z/16)
NS   115 K=Z-16*C
PK   120 IF K<=9 THEN T=K+48
SB   125 IF K>9 AND K<16 THEN T=K+55
QI   130 D(5-I)=T
OH   135 Z=C
FZ   140 NEXT I
QS   145 ? "w systemie szesnastkowym",
NH   150 FOR I=1 TO 4
CS   155 A$=CHR$(D(I))
HA   160 ? A$;
GS   165 NEXT I
     170 ?
     175 GOTO 15
QH   180 ? "WPROWADZILES LICZBE WIEKSZA OD
          NAJWIEKSZEJ LICZBY 16 BITOWEJ
     65535)"
 K   185 GOTO 15
[illegible]
[illegible]
[illegible]
```

PORTY JOYSTICKÓW (2)

W poprzedniej części artykułu zacząłem omawiać znaczenie poszczególnych bitów w rejestrze PACTL, podając znaczenie tylko drugiego bitu. Obecnie postaram się przybliżyć znaczenie pozostałych bitów.

Funkcje bitów w rejestrze PACTL są następujące:

bity 0—1 sterowanie CA1;
bit 2 gdy jest skasowany (0), to pod adresem PORTA jest rejestr porządkowania danych;
bity 3—5 sterowanie CA2;
bit 6 IRQA1, jeżeli podczas odczytywania rejestru PACTL bit 6 jest ustawiony (1), to znaczy, że wcześniej na CA2 zostało podane żądanie przerwania;
bit 7 IRQA2, tak jak bit 6, lecz sterowany przez CA1.

Po tym wstępie należy podać kilka wyjaśnień odnośnie CA1 i CA2. Otóż port PIA złożony jest z dwóch połówek, które posiadają dwa wejścia lub wyjścia kontrolne CA1 i CA2 (zależy to od ustawienia odpowiednich bitów w rejestrze PACTL). Złącze CA1 wykorzystywane jest w Atari jako źródło informacji dla systemu operacyjnego o przebiegu transmisji przez złącze szeregowe. Natomiast złącze CA2 wykorzystane jest do sterowania silnikiem magnetofonu. Zajmijmy się znaczeniem bitów i 1 i 0:

bit 1 bit 0
0 0 — przy opadającym zboczu sygnału CA1 bit 7 PACTL zostaje ustawiony (na 1). Wyjście przerwania A portu PIA pozostaje na "1";
0 1 — przy opadającym zboczu sygnału CA1 bit 7 PACTL zostaje ustawiony (1). Wyjście przerwania A portu PIA zmienia się na "0" i wywołuje żądanie przerwania z CPU;
1 0 — przy narastającym zboczu sygnału CA1 bit 7 PACTL zostaje ustawiony (1). Wyjście przerwania A portu PIA pozostaje na "1";
1 1 — przy narastającym zboczu sygnału CA1 bit 7 PACTL zostaje ustawiony (1). Wyjscie przerwania A portu PIA zmienia się na "0" i wywołuje żądanie przerwania z CPU.

Sterowanie CA2:

Gdy bit 5 PACTL jest skasowany (0), to CA2 służy jako wejście przerwań. Działanie opiera się na przedstawionej powyżej tabeli z tą różnicą, że funkcje są kontrolowane nie przez bity 0 i 2, lecz przez bity 3 i 4 rejestru PACTL, zamiast bitu 7 zmieniany jest stan bitu 6.

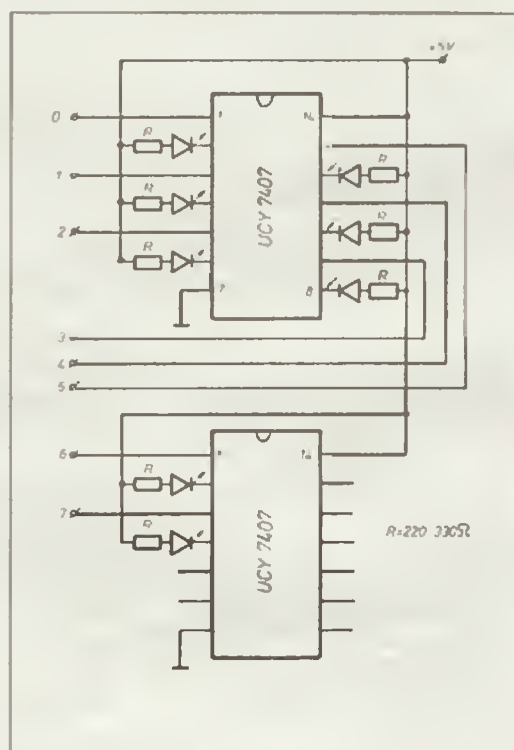
Gdy bit 5 PACTL jest ustawiony (1), to CA2 służy jako wyjście według poniższej tabeli:

bit 1 bit 0
0 0 — gdy bit 7 PACTL jest ustawiony przez CA1, to przy pierwszym ujemnym zboczu sygnału zegarowego po operacji odczytu z PORTA, CA2 zmienia stan na "0" i ponownie na "1";
0 1 — przy pierwszym ujemnym zboczu sygnału zegarowego po operacji odczytu z PORTA, CA2 zmienia stan na "0" i przy pierwszym ujemnym zboczu sygnału zegarowego, przy którym PIA nie odpowiada zmienia stan na "1";
1 0 — CA2 pozostaje na "0", dopóki nie zostanie zmieniona a kombinacja bitów;

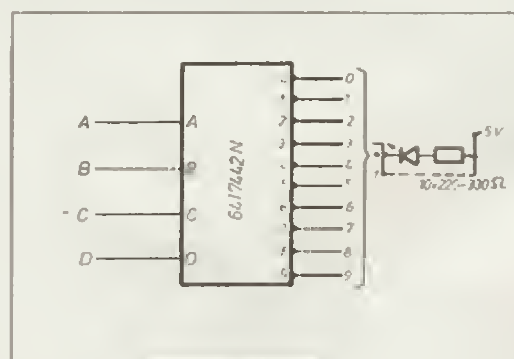
Po dokładnej analizie możliwości sterowania, znaczenie różnych zastosowań kanałów kontroli CA1 i CA2 staje się wyraźniejsze. Umożliwiają one komunikację PIA z różnymi układami peryferyjnymi, przy której CPU w dowolnej chwili otrzymuje sygnał przerwania, gdy dane wysłane z odbiornika są przetworzone lub nowe dane są gotowe do wczytania przez procesor. Istnieje przy tym możliwość wyłączenia przerwań, CPU może sprawdzać, czy są spełnione warunki przerwania. Przy wystąpieniu przerwania CPU może zlokalizować jego pochodzenie przez sprawdzenie bitów 6 i 7 PACTL. Można także wpływać na kanały kontroli bezpośrednio z urządzeń zewnętrznych, co jest znacznie szybsze niż sterowanie kanałów kontroli przez CPU. Pozwala to znacznie przyspieszyć szybkość transmisji danych oraz dokonać automatycznej synchronizacji między urządzeniami zewnętrznymi i systemem głównym.

Na razie tyle „mądrych" wiadomości, przejdźmy do układów praktycznych przedstawionych na rysunkach 1—5.

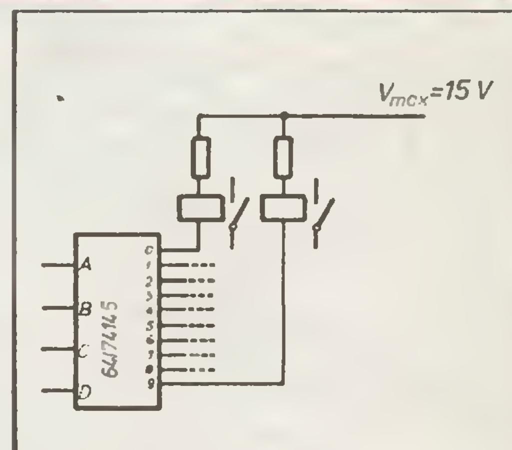
Krzysztof Kozłowski
Przemysław Strzelecki



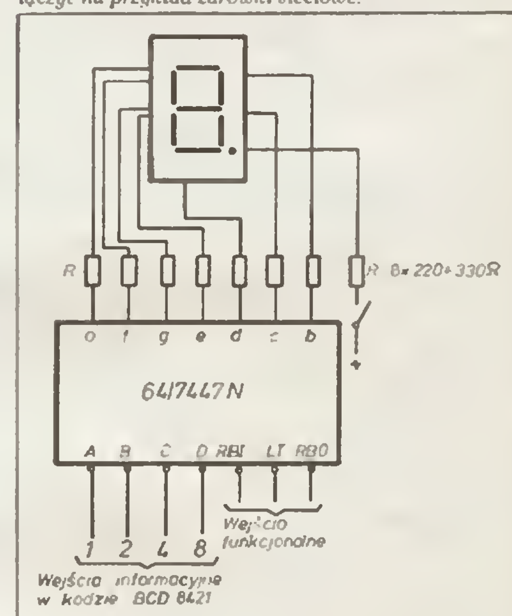
Rys. 1 — przedstawiono na nim schemat układu umożliwiającego uzyskanie dużej jasności świecenia diod. Wartości rezystorów R należy dobrać tak, aby uzyskać odpowiednią jasność, ale nie przekroczyć dopuszczalnej wartości prądu płynącego przez diodę. Uwaga ta odnosi się do wszystkich przedstawionych układów.



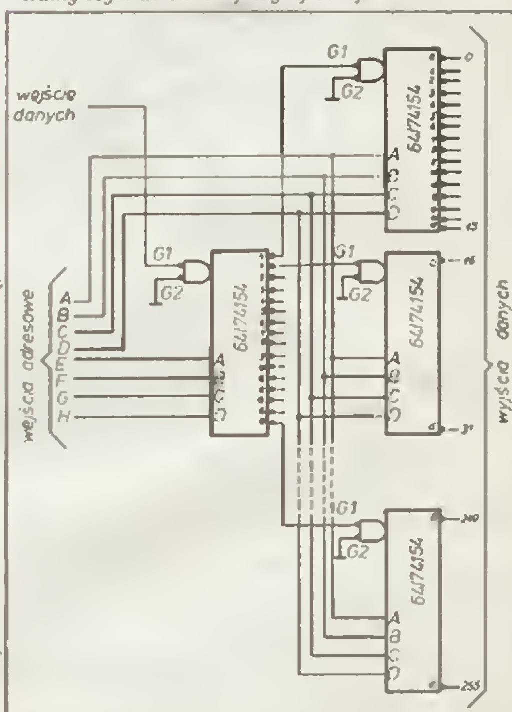
Rys. 2 — przy zastosowaniu układu UCY 7442 (dekoder 1 z 10) możliwe jest sekwencyjne sterowanie 10 diodami. Wykorzystując program przedstawiony w pierwszej części artykułu uzyskamy efekt przesuwającego się punktu świetlnego. Oczywiście zamiast diod można zastosować elektroniczne łączniki uruchamiające dowolne urządzenia. W następnej części artykułu postaram się przedstawić możliwości sterowania różnymi urządzeniami w zależności od stanu określonych czujników.



Rys. 3 — jest to w zasadzie taki sam układ jak opisany wyżej. Różnica polega jedynie na zastosowaniu dekodera z obwodem otwartego kolektora w stopniu wyjściowym. Dzięki temu możliwe jest dołączenie przekaźników na inne napięcie niż 5V, jednak nie wyższe niż 15V. Do styków przekaźników można dołączyć na przykład żarówki sieciowe.



Rys. 4 — jest to najpopularniejszy dekoder kodu binarnego na kod wskaźnika siedmiosegmentowego. Umożliwia on wzrokową obserwację aktualnej wartości pojawiającej się w wybranym porcie joysticków. Można do każdego portu dołączyć taki układ i stworzyć prosty program na przykład do podglądania zawartości określonych rejestrów, stoper czy programowalny zegar do ciemni fotograficznej.



Rys. 5 — przedstawiony schemat jest propozycją wykorzystania portów joysticków do sterowania 256 urządzeń. Bliższe informacje na temat tego układu, układów przedstawionych wcześniej oraz innych ciekawych rozwiązań można znaleźć w książce W. Sasala pt. „Układy scalone serii UCA64/UCY74, parametry i zastosowania", WKiŁ, 1985.

AUTOSTART PROGRAMÓW

Po opublikowaniu w „Bajtku" 10/87 odpowiedzi na list Arkadiusza Sochana otrzymaliśmy od Czytelników wiele listów. Kolega Marcin nie mając do czynienia na codzień z komputerami Atari udzielił niepełnego wyjaśnienia, co nasi Czytelnicy natychmiast zauważyli. Drukujemy więc dwa wybrane listy oraz komentarz redakcyjny.

BASIC ZX-SPECTRUM posiada instrukcję SAVE „nazwa" GOTO, która powoduje autostart programów. Nie wszyscy użytkownicy Atari widzą, że jego BASIC posiada identyczną instrukcję: SAVE "C:" (zapis) i Run "C:" (odczyt). W przypadku ATARI sprawa jest o tyle skomplikowana, że instrukcja SAVE"C:" daje sygnał o długich przerwach, co poważnie wydłuża transmisję danych. Można to ominąć używając krótkiego programu zapisanego instrukcją SAVE"C:" (tak zwanego Loadera) do ładowania i autostartu właściwego programu zapisanego w konwencji CSAVE — CLOAD.

Napiszmy więc odpowiedni Loader pamiętając jedynie, aby był on jak najkrotszy ze względu na format danych instrukcji SAVE"C:" i że po CLOAD sam się skasuje; n● tak! (litery przed nr linii to sumy kontrolne Edytora BASICa, nie należą do programu i nie należy je wpisywać).

```
FQ   10 ? CHR$(125):POKE 710,0:POKE 709,0:P
     OSITION 2,10:? "RUN":POSITION 0,0:POKE
      764,27:POKE 842,13:CLOAD
```

Teraz zapiszmy Loader na tasmie przed wlasciwym programem autostartującym instrukcją SAVE"C:" (tej taśmy nie przewijamy), a następnie wykonujemy NEW (czy wyłączamy komputer) i ładujemy właściwy program.
Dopisujemy do niego linię:
0 POKE 842,12:GR.0

Następnie zapiszemy, po programie Loader, program właściwy instrukcją CSAVE, najlepiej kasując ostatni sygnał programu ładującego Loader, jeżeli nie jest interpretowany (sprawdzcie, nie powinien).

Traktujcie program ładujący jak pilota do gry — nie powinno byc odstępu między pilotem a programem właściwym — a co najwyżej niewielki.
Teraz wyłączcie komputer i ładujcie całość RUN "C:"

Dlaczego działa?
Komputer wczytuje Loader, uruchamia go i dalej kolejno wykonuje:
1. *Czyszczenie ekranu*
2. *Czyni litery niewidzialnymi (POKE 710,0, POKE 709,0)*
3. *U dołu ekranu wpisuje RUN (tego nie widać, POS. 2, 10:PRINT „RUN")*
4. *Przenosi kursor do gory (POS.0,0)*
5. *Informuje komputer, że został naciśnięty jakiś klawisz (po CLOAD komputer nie czeka na to; POKE 764,27).*
6. *Uruchamia RETURN (POKE 842,13)*
7. *Wykonuje CLOAD*

Dowcip polega na tym, że RETURN jest wykonywane dopiero po zakończeniu programu, czyli po transmisji CLOAD. Wtedy RETURN dochodzi do RUN na ekranie i wykonuje go. Pierwsza linia programu przywraca stan normalny wyłączając RETURN i kasując ekran.

Jakub Cebula

Przedstawiony program jest tzw. „loaderem", który steruje ładowaniem programów i następnie uruchamia je. Program zajmuje obszar od adresu 1536 do 1751. Ponadto blokuje on klawisz RESET, ktory po wciśnięciu, kasuje pamięć RAM. Po dokładnym przepisaniu procedury, należy przygotować magnetofon do zapisu i uruchomić program rozkazem RUN. Zostanie on zapisany na taśmie w postaci BOOT, przeznaczonej do ładowania poprzez wciśnięcie klawisza START w momencie włączenia komputera.

Tomasz Waligóra

```
PR  1 REM ** (C) TOMASZ WALIGORA **
TN  2 REM *** 1987 WLOSZAKOWICE ***
AF  10 FOR G=0 TO 214:READ A:S=S+A
JH  20 POKE 1536+G,A:NEXT G
GN  30 IF S=13114 THEN 50
DY  40 ? "BLAD W DANYCH !":END
CF  50 OPEN #1,8,128,"C:"
UJ  60 POKE 850,11:POKE 852,0
AJ  70 POKE 853,6:POKE 856,215
KF  80 POKE 857,0:POKE 858,8
PE  90 A=USR(ADR("hhh LV "),16)
OM  95 CLOSE #1:? "O.K.":END
II  99 REM *** DANE ***
BB  100 DATA 0,2,0,6,16,6,169,60
PC  110 DATA 141,2,211,169,0,133,65
TI  120 DATA 96,169,148,141,197,2
MB  130 DATA 173,48,2,133,128,173
YR  140 DATA 49,2,133,129,160,4,177
CK  150 DATA 128,133,130,200,177
VH  160 DATA 128,133,131,230,131
HL  170 DATA 230,131,160,169,162,0
ID  180 DATA 189,91,6,145,130,232
YT  190 DATA 136,208,247,169,13,141
XL  200 DATA 74,3,169,0,133,84,169
XM  210 DATA 254,141,179,2,169,127
XJ  220 DATA 141,180,2,76,0,160,0,0
KS  230 DATA 0,0,0,0,0,0,0,0,0,0,0
KU  240 DATA 0,0,0,0,0,0,0,0,0,0,0
KW  250 DATA 0,0,0,0,0,0,0,0,0,0,0
KY  260 DATA 0,0,0,0,0,0,0,0,0,0,0
MP  270 DATA 0,9,9,4,0,0,56,8,50,36
ZC  280 DATA 33,8,50,51,53,29,56,26
LE  290 DATA 16,14,50,39,26,18,17
KT  300 DATA 12,20,22,23,0,37,43,47
CV  310 DATA 48,26,18,17,12,18,20
CR  320 DATA 24,0,37,43,0,0,47,48
EM  330 DATA 26,2,155,68,44,255,137
RL  340 DATA 155,150,0,68,137,40,52
XP  350 DATA 137,40,151,137,218,253
OQ  360 DATA 130,2,29,4,56,26,9,16
HB  370 DATA 20,8,4,56,0,45,41,36,0
EZ  380 DATA 0,0,0,0,0,0,0,0
```

Wszystkie rozwiązania tego problemu wymagają nagrania przed właściwym programem dodatkowego programu ładującego. Niestety, ominięcie tego jest niemożliwe. Najczęściej stosowane jest połączenie obu pokazanych wyżej sposobów, co pozwala na wyświetlenie tytułowej winiety programu już podczas jego wczytywania do komputera. Cały program składa się wówczas z trzech bloków: „loadera" w języku maszynowym (list 2), programu wyświetlającego tytuł i programu właściwego (oba nagrane przez CSAVE). Jednak procedura kolegi Jacka uniemożliwia to, gdyż wykorzystuje ekran do podania komputerowi rozkazów. W takim przypadku należy użyć instrukcji, którą już w 1983 roku opublikował amerykański miesięcznik „Antic".

```
PG  1090 POKE 65,0:POKE 764,32:X=USR(ADR("
    [illegible]"))
```

Zastępuje ona brakującą w Atari BASIC instrukcję CRUN.

Wojciech Zientara

Action! ZNACZY SZYBKOŚĆ



Często porównuje się szybkość pracy różnych języków programowania. Przyjmuje się przy tym przeważnie, że języki kompilowane są szybsze niż interpretowane. Jak jest rzeczywiście?

Przeprowadziłem test szybkości pięciu popularnych języków programowania. Jakie to języki? Przede wszystkim wbudowany na stałe do komputera Atari Basic. Ze względu na znaczną popularność uwzględniony został Turbo Basic XL. Jako trzeci język interpretowany został wybrany szeroko reklamowany Basic XE.

Z języków kompilowanych wybrałem C, który jest uważany za jeden z lepszych języków programowania, oraz dostępny jedynie dla komputerów Atari, bardzo popularny na Zachodzie język Action! Celowo został pominięty Pascal, gdyż jego wersja na Atari ma wszystkie procedury specyficzne dla tego typu komputera napisane bezpośrednio w języku maszynowym, co zniekształciłoby uzyskane wyniki. Dodatkowo wykonałem test programu w Atari Basic skompilowanego przy pomocy MMG Basic Compiler.

Do testu napisałem specjalny program. Zapełnia on najpierw punktami obraz w trybie wysokiej rozdzielczości (przy pomocy instrukcji graficznych), a następnie czyści go operując bezpośrednio na pamięci obrazu. Wydruki tego programu w pięciu wyżej wymienionych językach zamieszczone są obok.

Po tym wprowadzeniu czas na wyniki. Zawarte są one w poniższej tabelce. Mimo pominięcia Pascala oszacowałem także szybkość programu testowego w wersji Kyan Pascal. Program wykonywany jest przez około 8500 taktów zegara.

Język	liczba taktów
Atari Basic	24840
Basic skompilowany	5689
Basic XE	10807
Turbo Basic XL	9197
C (Deep Blue)	16709
Action!	4565

Zgodnie z oczekiwaniami najwolniejszy okazał się Atari Basic. Zaskoczeniem jest natomiast mała szybkość C — czas wykonywania programu (skompilowanego) był o 50 proc. dłuższy niż w interpretowanym Basicu XE i Turbo Basicu XL. Różnica pomiędzy tymi ostatnimi jest niewielka i praktycznie pomijalna. Stary Atari Basic pokazał jednak swoją wyższość po skompilowaniu — był prawie dwa razy szybszy od pozostałych Basiców.

Bezkonkurencyjny okazał się jednak Action! Nie ma żadnej przesady w twierdzeniu, że jest to najszybszy język Atari. W niektórych programach dorównuje on szybkością językowi maszynowemu. Z wydruku widać, że jest on składnią i strukturą zbliżony do Basica, nie ma więc kłopotów z przestawieniem się na inny sposób programowania przy przejściu z Basica na Action! Trzeba tu dodać, że Action! dysponuje jeszcze dodatkowymi procedurami operacji na blokach pamięci. Gdyby zostały one zastosowane w programie, to różnica szybkości byłaby jeszcze większa.

Wszystkich wątpiących w zalety Action! namawiam do samodzielnego wypróbowania tego języka, ja natomiast postaram się zamieścić wkrótce w „Bajtku" jakiś program w tym języku. Polecam także kurs programowania w Action! ukazujący się w odcinkach w „Bajtku-Atari".

Wojciech Zientara

```
;Test szybkosci - Action!

BYTE cl1=20,cl2=19,i
CARD j,time,sm,dl=560

PROC Test()
 cl1=0 cl2=0
 Graphics(24) Color=1
 FOR i=0 TO 191
  DO
  FOR j=0 TO 319
   DO
   Plot(j,i)
   OD
  OD
 sm=PeekC(dl+4) j=0
 WHILE j<7680
  DO
   Poke(sm+j,0)
   j==+1
  OD
 time=cl1+256*cl2
 Graphics(0)
 PrintE()
 Print("Czas wykonania testu: ")
 PrintC(time)
 PrintE(" taktow.")
RETURN
```

```
/* Test szybkosci - C */

char i,*cl1,*cl2;
int j,time,sm;

main()
$(
 cl1=20;
 cl2=19;
 *cl1=0;
 *cl2=0;
 graphics(24);
 color(1);
 for(i=0;i<=191;++i)
 $(
  for(j=0;j<=319;++j)
  $(
   plot(j,i);
  $)
 $)
 sm=dpeek(dpeek(560)+4);
 j=0;
 while(j<7680)
 $(
  poke(sm+j,0);
  ++j;
 $)
 time=*cl1+256*(*cl2);
 graphics(0);
 printf("\n");
 printf("Czas wykonania testu: ");
 printf("%d",time);
 printf(" taktow.\n");
$)
```

```
10 REM Test szybkosci - Basic
20 POKE 20,0:POKE 19,0
30 GRAPHICS 24:COLOR 1
40 FOR I=0 TO 191
50 FOR J=0 TO 319
60 PLOT J,I
70 NEXT J
80 NEXT I
90 DL=PEEK(560)+256*PEEK(561)
100 SM=PEEK(DL+4)+256*PEEK(DL+5):J=0
110 POKE SM+J,0
120 J=J+1
130 IF J<7680 THEN 110
140 TIME=PEEK(20)+256*PEEK(19)
150 GRAPHICS 0
160 PRINT
170 PRINT "Czas wykonania testu: ";
180 PRINT TIME;
190 PRINT " taktow."
200 END
```

```
10 REM Test szybkosci - Turbo Basic
20 TIME$= "000000"
30 GRAPHICS 24:COLOR 1
40 FOR I=0 TO 191
50   FOR J=0 TO 319
60     PLOT J,I
70   NEXT J
80 NEXT I
90 SM=DPEEK(DPEEK(560)+4):J=0
100 WHILE J<7680
110   POKE SM+J,0
120   J=J+1
130 WEND
140 J=TIME
150 GRAPHICS 0
160 PRINT
170 PRINT "Czas wykonania testu: ";
180 PRINT J;
190 PRINT " taktow."
200 END
```

```
10 Rem Test szybkosci - Basic XE
20 Dpoke 19,0
30 Graphics 24:Color 1
40 For I=0 To 191
50   For J=0 To 319
60     Plot J,I
70   Next J
80 Next I
90 Sm=Dpeek(Dpeek(560)+4):J=0
100 While J<7680
110   Poke Sm+J,0
120   J=J+1
130 Endwhile
140 Time=Peek(20)+256*Peek(19)
150 Graphics 0
160 Print
170 Print "Czas wykonania testu: ";
180 Print Time;
190 Print " taktów."
200 End
```

PROGRAM KOJĄCY NERWY (po raz drugi)

Ponieważ program opublikowany w numerze 1/88 wywołał bardzo duże zainteresowanie czytelników, powtarzamy go z myślą o pechowcach, którym nie udało się kupić styczniowego Bajtka. Program służy do sprawdzania poprawności wprowadzania do komputera wydruków zawierających „sumę kontrolną" w każdej z linii. Przypominam jednocześnie, że poleceniem |OFF można wyłączyć weryfikator i w dowolnym momencie włączyć ponownie poleceniem |ON.

(wz)

```
10  Weryfikator V.2 dla CPC 6128
20 WIDTH 40
30 MEMORY &A4FF
40 FOR a%=&A500 TO &A607
50 READ byte$
60 POKE a%,VAL("&"+byte$)
70 NEXT
80 CLS:PRINT
90 PRINT "Weryfikator V.2 zainst
alowany"
100 PRINT "Zaladuj swoj program"
110 PRINT "Rozkaz |CHECK,2 listu
je program"
120 PRINT "i sprawdza sume kontr
olna."
130 PRINT "Rozkaz |CHECK,8 listu
je program"
140 PRINT "z suma i drukuje na d
rukarce."
150 CALL &A500:|ON
160 DELETE
170 DATA 21,09,a5,01,0d,a5,c3,d1
180 DATA bc,00,00,00,00,18,a5,c3
190 DATA 2a,a5,c3,2f,a5,c3,43,a5
200 DATA 4f,ce,4f,46,c6,43,48,45
210 DATA 43,cb,00,00,cf,02,ac,c3
220 DATA a8,a5,21,27,a5,18,03,21
230 DATA 24,a5,28,06,cd,00,b9,c3
240 DATA 49,cb,11,5e,bd,01,03,00
250 DATA ed,b0,c9,4f,cd,00,b9,0d
260 DATA 28,08,0d,20,ea,dd,7e,02
270 DATA 18,04,7b,11,01,00,cd,a6
280 DATA c1,cd,64,e8,e5,4e,23,46
290 DATA 23,5e,23,56,e1,78,b1,c8
300 DATA cd,72,c4,e5,09,e3,cd,54
310 DATA e2,21,8a,ac,cd,7a,a5,e1
320 DATA 18,e2,e5,cd,ba,a5,e3,cd
330 DATA 98,a5,cd,53,f3,e3,cd,f6
340 DATA a5,cd,98,c3,e1,7e,a7,c8
350 DATA cd,98,a5,cd,98,c3,18,f5
360 DATA 3a,09,ac,d6,08,47,7e,a7
370 DATA c8,cd,1d,e2,23,10,f7,c9
380 DATA cd,24,a5,f5,c5,d5,e5,cd
390 DATA ba,a5,cd,f6,a5,e1,d1,c1
400 DATA f1,c9,eb,1b,af,47,67,6f
410 DATA 2f,32,23,a5,13,1a,d6,30
420 DATA 38,04,fe,0a,38,f6,1a,13
430 DATA a7,c8,4f,3a,23,a5,a1,fe
440 DATA 20,28,f3,79,fe,22,20,07
450 DATA 3a,23,a5,2f,32,23,a5,3a
460 DATA 23,a5,a7,79,c4,ab,ff,4f
470 DATA ad,07,6f,09,18,d8,3e,20
480 DATA cd,a0,c3,3e,5b,cd,a0,c3
490 DATA cd,44,ef,3e,5d,c3,a0,c3

Zmiany linii DATA dla CPC 464
240 DATA 43,cb,00,00,cf,98,aa,c3
270 DATA 06,dd,11,3a,bd,01,03,00
300 DATA 18,04,7b,11,01,00,cd,a2
310 DATA c1,cd,a3,e7,e5,4e,23,46
330 DATA cd,3c,c4,e5,09,e3,cd,63
340 DATA e1,21,a4,ac,cd,7a,a5,e1
360 DATA 98,a5,cd,96,f2,e3,cd,f6
370 DATA a5,cd,4e,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,4e,c3,18,f5
390 DATA 3a,24,ac,d6,08,47,7e,a7
400 DATA c8,cd,45,e1,23,10,f7,c9
490 DATA 23,a5,a7,79,c4,8a,ff,4f
510 DATA cd,5c,c3,3e,5b,cd,5c,c3
520 DATA cd,79,ee,3e,5d,c3,5c,c3

Zmiany linii DATA dla CPC 664
270 DATA 4c,cb,11,5b,bd,01,03,00
300 DATA 18,04,7b,11,01,00,cd,a9
310 DATA c1,cd,69,e8,e5,4e,23,46
330 DATA cd,75,c4,e5,09,e3,cd,59
360 DATA 98,a5,cd,58,f3,e3,cd,f6
370 DATA a5,cd,9b,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,9b,c3,18,f5
400 DATA c8,cd,22,e2,23,10,f7,c9
510 DATA cd,a3,c3,3e,5b,cd,a3,c3
520 DATA cd,49,ef,3e,5d,c3,a3,c3
```

Ach, co to był za błąd?!

Doświadczeni programiści wiedzą, że aby znaleźć odpowiedź na takie pytanie trzeba niejednokrotnie poświęcić dużo czasu, chociaż odpowiedź może okazać się bardzo prosta.

W dużo gorszej sytuacji są początkujący amatorzy „komputerowego szaleństwa", stawiający pierwsze kroki nie tylko w programowaniu, ale również i w nieodzownym przy tej pracy języku angielskim. Krótkie "syntax error" lub inne niezrozumiałe komunikaty rujnują zapał, nerwy, i czupryny naszych najmłodszych czytelników. A właściwie jest to już czas przeszły... Wystarczy uzbroić się w odrobinę cierpliwości i przepisać poniższy program oraz utrwalić go na kasecie lub dyskietce pod nazwą „SUFLER". Od tej pory pisanie własnych programów należy rozpoczynać od załadowania naszego „suflera", numerując własne linie począwszy od 10. i kończąc **obowiązkowo** linią z rozkazem END o numerze nie większym niż 59999! Każde uruchomienie programu z błędem spowoduje pojawienie się informacji o numerze błędu, numerze linii, w której wystąpił błąd oraz komunikatu w języku angielskim z następującym po nim obszerniejszym wyjaśnieniem w języku polskim.

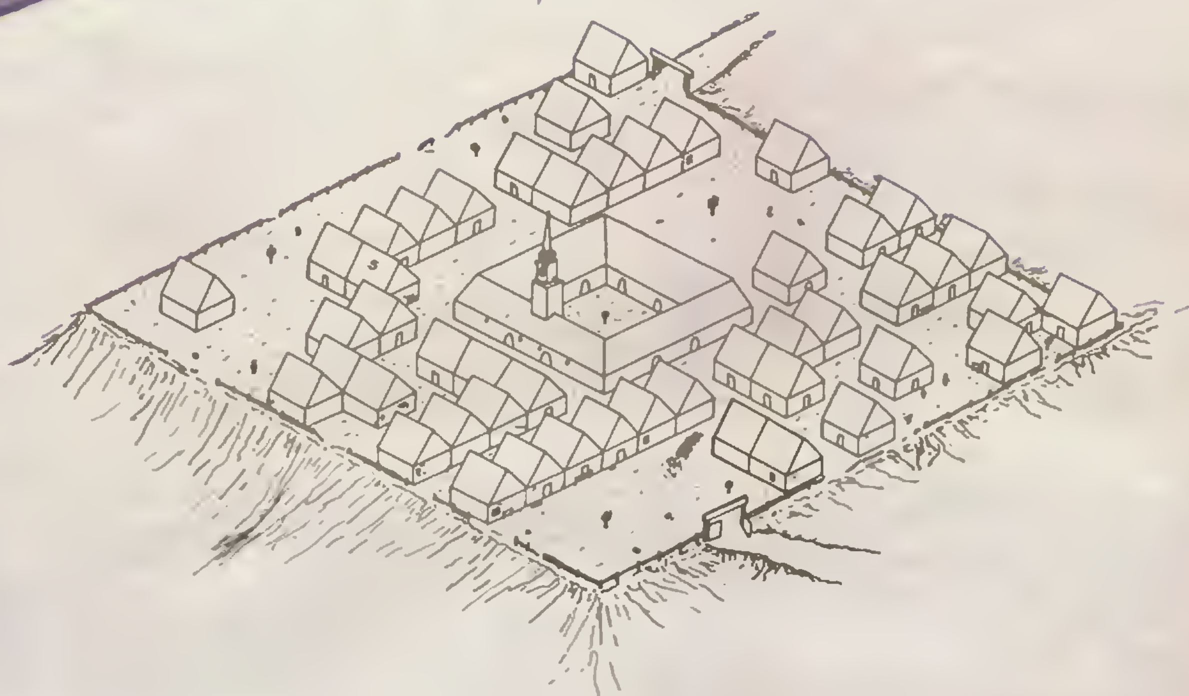
Sprawdzony i działający program zapisuje się oczywiście na kasecie lub dyskietce po uprzednim usunięciu „duflera" poleceniami DEL — 9 i DEL 60000 —. Zapisane wcześniej a nie działające programy mogą być poprawiane również przy użyciu tego programu po włączeniu do nich „suflera" poleceniem CHAIN MERGE „nazwa programu" (pod warunkiem wszakże, że kończą się również linią z poleceniem END).

Roztargnionym przypominam, iż nie należy przepisywać z listingu liczb podanych w nawiasach kwadratowych, ani samych nawiasów. Są to liczby kontrolne, generowane przez opisywany poniżej po raz drugi weryfikator, służące wyłącznie do kontroli poprawności wpisywanych linii programu

Wojciech Ziółek

```
1 ON ERROR GOTO 60000                [1511]
2 'Od tego miejsca mozesz             [609]
3 'wprowadzac swoj program            [1654]
4 'az do linii nr 59999               [656]
60000 CLS:PRINT "Blad nr ";ERR;" [2899]
 w linii nr";ERL
60010 ON ERR GOTO 65000,65010,65 [10122]
020,65030,65040,65050,65060,6507
0,65080,65090,65100,65110,65120,
65130,65140,65150,65160,65170,65
180,65190,65200,65210,65220,6523
0,65240,65250.65260,65270,65280,
65290,65300,65310
65000 PRINT:PRINT " Unexpected N [13481]
ext : Proba wykonania instrukcji
 NEXT, dla ktorej brak odpowiada
jacej jej instrukcji FOR lub zmi
enna wystepujaca po NEXT jest in
na niz w instrukcji FOR":END
65010 PRINT:PRINT " Syntax Error [7377]
 : Skladnia polecenia niezrozumi
ala dla Basic'a":END
65020 PRINT:PRINT " Unexpected R [14153]
eturn : Proba wykonania instrukc
ji RETURN, dla ktorej nie bylo w
ywolania podprogramu instrukcja
GOSUB":END
65030 PRINT:PRINT " Data Exauste [8321]
d : Brak lub za malo danych dla
instrukcji READ":END
65040 PRINT:PRINT " Improper Arg [16123]
ument : Ogolny komunikat o bledz
ie - bark zgodnosci argumentow w
ykonywanej funkcji z jej paramet
rami lub bledne parametry polece
nia":END
65050 PRINT:PRINT " Overflow : W [16600]
ynik operacji arytmetycznych prz
ekracza dopuszczalny zakres licz
b rzeczywistych lub przy zamiani
e dopuszczalny zakres liczb calk
owitych":END
65060 PRINT:PRINT " Memory Full  [7785]
 : Za duzy program lub zbyt wiele
 zmiennych lub zbyt wiele petl:
typu GOSUB,WHILE i FOR":END
65070 PRINT:PRINT " Line Does no [6121]
t Exist : Brak linii o takim num
erze":END
65080 PRINT:PRINT " Subscript Ou [8626]
t of Range : Jeden ze wskaznikow
 zadeklarowanej tablicy jest za
duzy lub za maly":END
65090 PRINT:PRINT " Array Alread [7712]
y Dimensioned : Zadeklarowanie p
o raz drugi tej samej tablicy":E
ND
65100 PRINT:PRINT " Division By  [13361]
Zero : Jedna ze zmiennych w lini
i wykonujacej dzielenie rowna je
st zeru lub wynik przekracza zak
res wartosci liczbowych":END
65110 PRINT:PRINT " Invalid Dire [9880]
ct Command : Polecenie nie moze
byc wykonane w trybie bezposredn
im":END
65120 PRINT:PRINT " Type Missmatc [18699]
h : Zamiast zmiennej numerycznej
 wprowadzona zmienna alfanumeryc
zna (lub odwrotnie) lub blad prz
y wprowadzaniu wartosci dla inst
rukcji READ lub INPUT":END
65130 PRINT:PRINT " String Space [10610]
 Full : Nie ma juz miejsca na pr
zechowywanie wyrazen lancuchowyc
h":END
65140 PRINT:PRINT " String Too L [6222]
ong : Lancuch znakowy przekracza
 255 znakow":END
65150 PRINT:PRINT " String Expre [12653]
ssion Too Complex : Zbyt zlozone
 wyrazenie tekstowe dajace przy
manipulacji nim za duzo posredni
ch wartosci tekstowych":END
63160 PRINT:PRINT " Cannot CONT( [12429]
inue) : Nie mozna wznowic dziala
nia programu przez uzycie CONT,
poniewaz zmodyfikowano jego tres
c":END
65170 PRINT:PRINT " Unknown User [10654]
 Function : Proba uzycia funkcji
 FN uzytkownika bez jej uprzedni
ego zdefiniowania (DEF FN)":END
65180 PRINT:PRINT " RESUME missi [7473]
ng : Brak instrukcji RESUME w pr
ogramie obslugi bledow":END
65190 PRINT:PRINT " Unexpected R [6122]
ESUME : Program zawiera instrukc
je RESUME bez poprzedniego wpisa
nia ON ERROR GOTO...":END
65200 PRINT:PRINT " Direct Comma [13461]
nd Found : Podczas ladowania pro
gramu wystapila linia bez numeru
 traktowana jako zlecenie bezpos
rednie":END
65210 PRINT:PRINT " Operand Miss [7082]
ing : Brak fragmentu polecenia,z
la skladnia":END
65220 PRINT:PRINT " Line Too Lon [5045]
g : Linia za dluga (>255 znakow)
":END
65230 PRINT:PRINT " EOF Met : Pr [10057]
oba odczytu nastepnej informacji
 po napotkaniu znacznika konca z
bioru":END
65240 PRINT:PRINT " File Type Er [9664]
ror : Proba wykorzystania zbioru
 o innym niz dozwolony typie":EN
D
65250 PRINT:PRINT " NEXT Missing [4661]
 : Brak instrukcji NEXT zamykaja
ca petle rozpoczeta przez FOR":E
ND
65260 PRINT:PRINT " File Alraedy [10520]
 Open : Proba otwarcia kolejnego
 zbioru bez zamkniecia poprzedni
ego":END
65270 PRINT:PRINT " Unknown Comm [7041]
and : nieznane polecenie lub slo
wo kluczowe":END
65280 PRINT:PRINT " WEND Missing [7279]
 : Brak instrukcji WEND zamykaja
cej petle, rozpoczyajaca sie od
WHILE...":END
65290 PRINT:PRINT " Unexpected W [8405]
END : Napotkanie instrukcji WEND
 , dla ktorej brak odpowiadajacej
 jej instrukcji WHILE":END
65300 PRINT:PRINT " File Not Ope [7622]
n : Proba uzyskania dostepu do z
bioru, ktory nie zostal otwarty"
:END
65310 PRINT:PRINT " Broken In :  [3564]
Przerwanie":END
```

GUN



Następna gra firmy Ultimate — GUN FRIGHT, jest bardzo podobna do NIGHT SHADE. Tu jednak nie ma mowy o duchach — wrogiem do zlokalizowania i unieszkodliwienia jest materialna postac. Chodzi w kapeluszu i zawsze z ręką na colcie. Jest jednym z najsławniejszych i najszybszych łatrów Dzikiego Zachodu. Do szajki należą Buffalo Bill, Billy Kid, Sundance Kid, Butch Casidy, Ma Baker, Wild Bill, Rumpo Kid, Clever Jake, i wielu innych. Ty, jako szeryf miasteczka Draw City musisz ich wszystkich posłać do piachu, gdyż nie dadzą Ci spokoju. Codziennie rozlepiasz na ścianach listy gończe, ofiarowując coraz większą nagrodę za schwytanie bandyty. Niestety, mieszkańcy Draw City nie są kapusiami, tylko mali chlopcy pokazują Ci palcem, w którą stronę poszedł poszukiwany morderca.

Gdy jako szeryf Quickdraw maszerujesz ulicami miasteczka, ściany budynków zastępowane są liniami, byś mógł cały czas obserwowac bohatera. Twoje uzbrojenie to szescIostrzalowy colt Magnum 0,72 mm. Po wystrzeleniu całego magazynka broń samoczynnie się laduje, odliczając jednak od posiadanej przez Ciebie kwoty sumę potrzebną na zapłacenie za naboje. Aby zdobyć pieniądze, na początku gry strzelasz do spadających worków ze złotem. Jest to dobry trening przed starciem z poszukiwanymi.

Przez cały czas z lewej strony ekranu widzisz list gończy z wizerunkiem poszukiwanego bandyty i cenę za jego głowę. Pod tym plakatem wisi kilka kapeluszy. Jest to liczba ewentualnych przegranych pojedynków, jaka Ci pozostała. Jeszcze niżej pokazany jest magazynek colta i ilość gotówki, jaką dysponujesz oraz ceny naboi, koni i grzywna za zabicie niewinnego mieszkańca miasteczka.

...RIGHT

...rzywna ta jest bardzo wysoka — od ...00 do 5000 dolarów. Czasem jednak ...ie ma wyjścia. Gdy ktorys z mieszkan...ów wejdzie Ci w drogę, musisz wybrac ... albo Ty, albo on.

Na ulicy można znalezc konia na ...woch nogach. Po dotknięciu go i odli...zeniu sumy na zapłacenie, galopujesz ...o mieście bez obawy przed jego miesz...ancami. Czasami i bandyta jest konno, ...ięc uważaj, bo może Ci uciec.

Jeśli całą forsę przepuścisz na nabo..., konie i kary, nie martw się, tylko do...rze rozejrzyj po okolicy. Na pewno znaj...ziesz porzucone worki ze złotem.

Gdy bandyta znajdzie się w zasięgu ...wojego wzroku, od razu strzelaj, gdyż ...st on bardzo szybki i celny. Sam poje...ynek odbywa się w pobliskim barze. ...rzed sobą widzisz mordercę i swoj ce...wnik. Strzelaj bez namysłu. Zanim ...andyta wyciągnie rewolwer, minie jed...a sekunda, przez ten czas musisz go ...efic chociaż w nogę. Jesli nie zdążysz, ...n podziurawi Twój życiorys odejmując ... jeden kapelusz. Jeżeli jednak szybko...cią strzałow i ich celnością poślesz do ...emi mordercę, nie ciesz się za bardzo, ...to kolejny bandyta zaczyna rozrabiać, ...obie przypada unieszkodliwienie go; ...czywiście stawka jest już większa. I je...cze jedna uwaga: podczas penetro...ania miasta uważaj na kaktusy, gdyż ...otknięcie grozi śmiercią i tylko leżący ... ziemi kapelusz będzie pamiątką po ...euważnym szeryfie. Zwrócić też uwa...ę na roj dzikich szerszeni, ktory wypu...czają żądni zemsty kumple zabitych ...rów.

Gra nigdy się nie konczy (chyba, że ...zegrasz), gdyż ciągle rodzą się nowi ...andyci i wciąż trzeba dbać o porządek ... mieście.

...rma: Ultimate

...omputer: ZX Spectrum 48/+, Commodore 64/128.

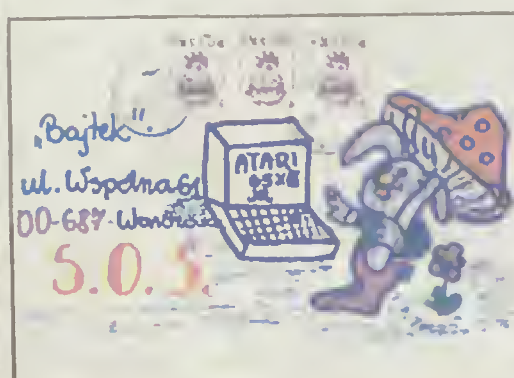


10 BAJTKOWA LISTA PRZEBOJÓW (4/88)

To już 25-te notowanie Bajtkowej Listy Przebojów Zainteresowanie nią nie słabnie. Wciąz otrzymujemy od Was setki listów z propozycjami — w tym miesiącu nadeszło ich 1750. Głosowano na 194 tytuły.

			ATARI	AMSTRAD	COMMODORE	SPECTRUM
1	SABOTEUR II	↑		×	×	×
2	TRASHMAN	↓		×		×
3	EQUINOX			×	×	×
4	GUN FRIGHT			×	×	×
5	ARKANOID	↑	×	×	×	×
6	CHIMERA	!	×	×	×	×
7	MIKIE	↑		×	×	×
8	DAM BUSTERS	↓		×	×	×
9	BARBARIAN	↓	×	×	×	×
10	EXOLON	!			×	×

Nagrody wylosowali Małgorzata Łącka i Marcin Jarek.



CO JEST GRANE



TRASHMAN

Czy pamiętasz Paper — boy'a? Chyba pomagałeś mu rozwozić gazety do londyńskich domów? Nasz kolega nie chodził do szkoły, tylko uczył się zarabiać na własne utrzymanie. Skutki tego widać do dzisiaj — ma rodzinę i zajmuje się... zbieraniem śmieci. Przechodzi codziennie tę samą ulicę i opróżnia kosze usytuowane koło domów. Nie potrzeba wiele wiedzy, prawda? Jednak liczy się szybkość i zręczność! Trashman wie, że nawet w tym fachu może zrobić karierę, może również sporo zarobić. Opanował sposoby zdobywania napiwków, potrafi też pracować na tyle szybko, by zostać przeniesionym na bardziej elegancką i lepiej płatną ulicę. Ty zapewne nie dorobiłeś się jeszcze trzech liter przed nazwiskiem (jeśli tak, może się Pan zawsze wytłumaczyć chęcią zabawy) — chwyć więc za joystick i zdobądź tytuł trashmana roku! Zaczynamy od małej Montague Road.

Jesteś tu po raz pierwszy? To zrozumiałe, ale liczy się szybka orientacja. Przez furtkę do domu po lewej stronie, kosz stoi z drugiej strony. Nie po trawniku! Angielskie trawniki są elastyczne i wytrzymałe, ale nie denerwuj gospodarza. Teraz z powrotem tą samą drogą do samochodu, który przystanął przy krawężniku. Podejdź do niego od tyłu, a gdy kosz zostanie opróżniony, odnieś go na swoje miejsce. I szybko na drugą stronę ulicy. Stop! To jest Londyn, uważaj na samochody! Z drugim koszem ta sama operacja. Jeśli nie widać ścieżki, idź blisko ściany budynku. Potem szybko w górę ekranu do następnego domu. Masz już kosz w ręku, ale samochód ucieka Ci, nie możesz wcelować w furtkę i... Miało być szybko, a tu: bonus — 0, joystick nie reaguje, a na środku ekranu słodka reprymenda: You're too slow" — jeszcze możesz spróbować, ale uważaj, żebyś nie musiał szukać innej pracy.

Teraz w duchu postanawiasz, spokojnie, bez nerwów, od domu do domu. Zaczynamy... Pierwszy kosz... dobrze... odnosimy, z powrotem... Och! Pisk opon i... oczom Twoim ukazuje się krótka notatka na czarnym tle — Trashman... zostawił żonę i troje dzieci oraz 100 zdobytych dotychczas punktów. Dobrze, że to tylko gra.

Zauważyłeś zapewne, iż po bezbłędnym obsłużeniu posesji, gospodarz pojawia się w drzwiach i proponuje Ci napiwek lub zarobienie czegoś „na boku". Warto skorzystać. Kilka sekund spędzisz w budynku, ale bonus i tak się powiększy.

Na pewno po kilku próbach opróżnisz pięć koszy na Montague Road. Zostaniesz wtedy przeniesiony na Pulteney Road, ulicę dłuższą i trudniejszą do przejścia, ale zamieszkałą przez hojniejszych londyńczyków. Mają oni bardziej rozbudowane posiadłości, ich dzieci bardzo lubią jeździć na rowerach, a pieski podwórzowe dbają o ogródki. Mimo to powinieneś również tam ukończyć swą pracę w wyznaczonym czasie i zostać przeniesionym na jedną z głównych ulic dzielnicy — Cro... Nie! Przekonaj się sam, ile ulic jest w Londynie.

Gra ma jeszcze jedną zaletę. Nawet gdy Twoja znajomość języka angielskiego nie jest doskonała, na pewno ubawią Cię komentarze pojawiające się na ekranie. Dowiesz się na przykład o człowieku, który kazał wyrzucić komputer swego syna oraz o nowym modelu Spectrum 4 MB. Warto zagrać!

Firma: New Generation Softvare
Komputer: ZX Spectrum 48/+ Commodore 64/128, Amstrad/Schneider.

(pb)



SHOW JUMPING

Na pewno oglądałeś nieraz w telewizji lub z trybun zawody hippiczne. Biorą w nich udział najlepsi jeźdźcy w kraju. Jeśli jednak i ty chcesz spróbować swych sił w tej dyscyplinie sportu, zagraj w Show Jumping.

Jesteś dżokejem biorącym udział w corocznie organizowanych zawodach jeździeckich. Do przejechania masz tor składający się z pięciu do piętnastu przeszkód. Są to murki, rowy z wodą, poprzeczki na stojakach, poziome żerdzie, beczki. Za strącenie przeszkody otrzymujesz odpowiednią ilość punktów karnych. Gdy ilość punktów karnych przekroczy dwadzieścia zostajesz zdyskwalifikowany. Dyskwalifikacja następuje również, gdy twój koń odmówi posłuszeństwa trzy razy.

W zachowaniu odpowiedniej kolejności przeskakiwania pomogą Ci litery umieszczone przy każdej przeszkodzie i jeździec na koniu w górnej części ekranu pokazujący kierunek skoku przez przeszkodę.

Ilość punktów karnych przedstawiona jest w lewym górnym rogu ekranu, a czas w prawym.

Przed rozpoczęciem gry możesz wybrać ilość graczy oraz konia, na którym chcesz jechać. Gdy wybierzesz, zobaczysz plan toru, na którym są zaznaczone kierunki pokonywania przeszkód, start i meta. Poziom trudności gry wybierasz na początku. Wybór jest prosty: hard (trudno) lub easy (łatwo). Sterowanie: Kempston joystick, Sinclair joystick, keyboard.

W grze może brać udział do ośmiu graczy, którzy mają do wyboru po jednym z ośmiu koni. Po zakończeniu konkurencji wyświetlana jest tablica wyników wraz z czasami przejazdów i ilościami punktów karnych (foul) dla każdego zawodnika.

Grę cechuje dokładna, staranna grafika i niezłe, choć mało zróżnicowane efekty dźwiękowe. W sumie jest to ciekawa i dobra zabawa do wykorzystania w rodzinnym gronie podczas długich, nudnych wieczorów.

Autor: Elliot Gay
Komputer: ZX Spectrum 48/+, Commodore 64/128, Amstrad/Schneider.

(mateo)

KRÓL I KRÓLOWA GIER



Małgorzata Łącka lat 9, uczennica I klasy Szkoły Podstawowej nr 164 w Warszawie. Gra (na Spectrum 48) najczęściej z bratem, Grzegorzem, lecz jak twierdzą zgodnie oboje, jest od niego znacznie lepsza. Hobby: książki.



Marcin Jarek lat 12, uczeń klasy Vb Szkoły Podstawowej nr 57 w Warszawie jest bardzo dobrym uczniem. Lubi grać na swoim Spectrum Plus. Ulubiona gra: TURBO. Hobby: sport, w szczególności jazda na rowerze.

S.O.S.

Szukam instrukcji do gry TIGERS IN THE SNOW na ATARI z magnetofonem oraz instrukcji do SPY VS SPY II na ten komputer. Pragnę wymienić oprogramowanie. Odstąpię instrukcję SILENT SERVICE.

Piotr Kołuda
ul. Limanowskiego 154 m 43
91-032 Łódź

Mam komputer ATARI 800 XE. Proszę o podanie nieśmiertelności do gier: PRELIMINARY MOUNTY, SHAMUS, ZONE X.

Wojciech Franaszczuk
ul. Wrocławska 19
55-050 Sobótka

Proszę o plan i nieśmiertelność do gry THE GREMLINS na ATARI.

Krzysztof Zedlewski
ul. Brzozowa 13
82-200 Malbork

Jak dojść w grobowcu do drzwi z podkową, butem i trąbką? Jak uwolnić księżniczkę w grze ZORRO w wersji na ATARI? Nie mogę sobie także poradzić z kodem w grze PHARAOHS CURSE. Poszukuję też dokładnego opisu do gry SWAT.

Andrzej Zasłonka
ul. Piotrkowska 182 m 138
90-368 Łódź
telefon 36-36-36

Zwracam się o pomoc do czytelników: szukam mapy do gry ICE PLACE, a także nieśmiertelności do gier: BOULDER DASH III, LAZY JONES i BEACH HEAD na Commodore 64.

Sebastian Cyrson
ul. PCK 5 m 86
81-621 Gdynia

Nie wiem, jak przejść planszę „g" w grze BOULDER DASH. W grze GREEN BERET poszukuję nieśmiertelności. Mam komputer ATARI 130 XE.

Wojciech Glanowski
Os. Skalska 2 m 22
32-340 Wolbrom

Moją ulubioną grą jest KILLER GORILLA. Niestety, nie wiem, jak przejść trzecią planszę — nie umiem wskoczyć na windę.

Maciej Zieliński
ul. M. Nowotki 44 m 112
32-300 Olkusz

Jak uzyskać nieśmiertelność w grach: BRUCE LEE, MONTEZUMA'S REV., CRYSTAL RAIDER w wersji kasetowej na ATARI 800 XL?

Patryk Zawadowicz
ul. Matejki 40a m 2
72-600 Świnoujście

Poszukuję instrukcji do gry SILENT SERVICE. W zamian służę opisami do gry F-15 i KENNEDY APPROACH.

Jakub Lercel
ul. Dobrego Pasterza 125 m 87
31-416 Kraków

Pomóżcie! Kto mi przyśle opisy do gier: STORM, BILBO, PHANTOM na komputer ATARI?

Daniel Chojecki
ul. Lekarska 16 m 16
00-610 Warszawa

Proszę o pomoc w grze RAMBO. Umiem zebrać wszystkie rodzaje broni, a dalej nie wiem, co robić.

Fryderyk Badura
ul. Dworcowa 5 m 41
44-200 Rybnik

Mam komputer ATARI 800 XL. Proszę o pomoc w grach NEW YORK CITY i KING OF RING.

Radosław Forma
ul. Kościuszki 19 m 80
41-300 Dąbrowa Górnicza

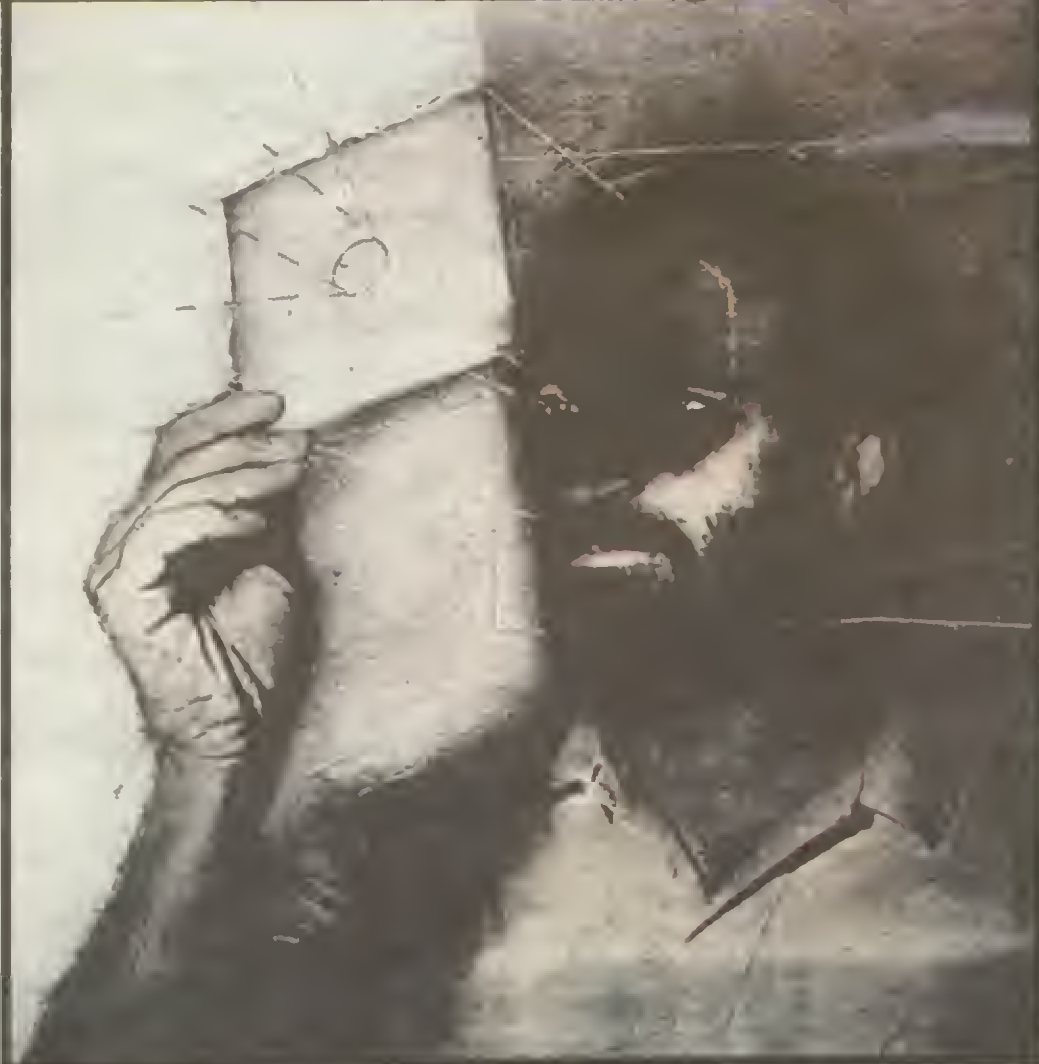
W grze APOLLO MISSION dochodzę do drugiej planszy i nie umiem przejść dalej. Proszę także o adres firmy wysyłającej programy pocztą po niskich cenach. Mam Commodore +4.

Artur Szubert
ul. Raciborska 2
47-480 Pietrowice Wielkie

Może ktoś mi pomoże, nie wiem jak grać w grę EXORCIST oraz jak przejść planszę w trzecim etapie gry CAVE FIGHTER, obie w wersji na C-16.

Jarosław Wierny
ul. Zubrzyckiego 25b m 207
45-263 Opole

ALGORYTM HORNERA



Od dawna już nie mieliśmy w „Następnym Kroku” okazji programować. Żebyście przypadkiem nie nabrali podejrzeń, że informatyka to dziedzina bardziej teoretyczna niż praktyczna, dziś zadanie praktyczne — będziemy obliczać wartość wielomianu.

Zeby wszyscy mieli równe szanse zaczynamy od podstaw. Wielomian jest to funkcja zapisana następującym ogólnym wzorem:

$$a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_2 x^2 + a_1 x + a_0$$

Liczba n jest stopniem wielomianu. Wielomian stopnia n ma n+1 współczynników (liczby a_n, a_{n-1}, ... a_2, a_1, a_0). Kazdy z tych współczynników jest pomnozony przez odpowiednią potęgę zmiennej x. Oczywiście podany wzór jest to ogólny wzór na wielomian. Konkretny wielomian otrzymamy ustalając stopień, np. n=4 i podając współczynniki (4+1=5 liczb) a_4 do a_0. Np dla $a_4=7$, $a_3=-2$, $a_2=1$, $a_1=2$, $a_0=11$, otrzymujemy następujący wielomian stopnia czwartego:

$$7x^4 - 2x^3 + x^2 + 2x + 11$$

Dodajmy jeszcze, ze wielomian stopnia pierwszego, to zwykła funkcja liniowa (jej wykresem jest prosta), a stopnia drugiego to trójmian kwadratowy, prezentowany w szkole średniej (jego wykresem jest parabola).

Oczywiście do konkretnego wielomianu możemy wstawiać różne wartosci zmiennej x. Otrzymamy wtedy różne wartości funkcji. Obliczenie tych wartości gdy znamy wartości współczynników oraz zmiennej x nie jest zadaniem wymagającym wiele myslenia. Trzeba po prostu wykonać odpowiednie operacje potęgowania, mnożenia potęg x przez współczynniki oraz dodawania otrzymanych iloczynów.

Proste, prawda? Możemy juz chyba z łatwością pisać program, który zrobi to za nas. Ale przecież miało być od podstaw, więc jak mamy zapisać wielomian w pamięci komputera? Stopień zapiszemy w zmiennej, najlepiej nazwać ją n, zaś do zapisania współczynników użyjemy tablicy jednowymiarowej, mającej n+1 elementów — liczb rzeczywistych. W jednym elemencie zapiszemy jeden współczynnik: w A(0) – a_0, w A (1) – a_1, itd. W BASIC-u deklaracja potrzebnej tablicy mogła by wyglądać:

```
DIM A(N)
```

W Pascalu zaś:

```
A:arrary[0..n] of real;
```

Niestety jednak tak zadeklarować tablicy nie możemy — jej wymiar nie może być zmienną, nie pozwalają na to reguły języka programowania (w Pascalu taka konstrukcja jest w ogóle niedozwolona, w BASIC-u nie może wystąpić wewnątrz pętli) Co więc zrobić? Deklarować: array[0..5] gdy chcemy liczyć wielomian stopnia 5 i zmieniać deklarację na array 0..100 gdy n=100? To bez sensu — dla każdej zmiany stopnia trzeba nowy program!

Zrobimy inaczej, zadeklarujemy tablicę tak długą, aby zmieścił się w niej każdy z przewidywanych wielomianow, a następnie dla konkretnego n będziemy wykorzystywać początkowy fragment, czyli elementy od A(0) do A(n).

Jeśli tablica A zawiera współczynniki wielomianu, to możemy obliczyć jego wartość w punkcie x jedną, prostą pętlą:

```
30 W=0
40 FOR I=0 TO N
50 W=W+A(I)*X↑I
60 NEXT I
70 REM Tu W zawiera obliczoną wartość wielomia-
   nu.
```

Spróbujmy teraz zapisać to samo w Pascalu. Ponieważ jednak nie ma w nim operacji potęgowania musimy zrobić to nieco inaczej. Zauważmy, ze po obliczeniu wartości x^i, aby obliczyć x^{i+1} nie potrzebujemy wcale używać potęgowania, wystarczy mnożenie x^i*x. Inaczej mówiąc, zaczniemy od znanej wartości x (pamiętamy, że $x=x^1$), x^2 uzyskamy jako x^1*x, $x^3=x^2*x$, $x^4=x^3*x$, itd. Wykorzystamy to w poniższym programie Zmienna xp będzie zawierała wartości kolejnych potęg x

```
program wielomian (input, output);
{program obliczania wartości wielomianu}
var
x,xp,w: real;
n,i    :integer;
A:array [0..100] of real;
begin
writeln ('Podaj stopień n'),
readln(n);
(generacja współczynników wielomianu zamiast ich
wczytywania — przyspieszy testowanie programu)
for i: = 0 to n do A[i]:=i;
writeln ('Podaj x');
readln (x);
(zaczynamy obliczenia)
w:=A[0];
xp:=1;
for i:=1 to n do
begin
xp:=xp*x;
w:=w+A[i]*xp:
end;
writeln ('Dla x=', x,' wartość wielomianu wynosi',
w)
end
```

Mamy dwa różne programy, wykorzystujące dwie metody obliczenia tej samej wartości Jak porównać szybkość ich działania? Oczywiście najprościej zrobimy to uruchamiając każdy z nich i mierząc czas wykonania. Trzeba jednak pamiętać o kilku istotnych sprawach:

- oba programy muszą być napisane w tym samym języku i tlumaczone tym samym kompilatorem (interpreterem) — chodzi nam o porównanie szybkości programów a nie translatorów;
- należy mierzyć tylko czas wykonywania obliczeń — bez wczytywania danych itd.;
- drobne różnice mogą być trudne do uchwycenia przy pomiarze czasu zegarkiem czy stoperem, wtedy fragment programu wykonujący obliczenia można umieścić we wnętrzu pętli, która wykonuje się np 10 lub 100 razy, odpowiednio zwielokratniając różnicę i ułatwiając pomiar

Zachęcam do przepisania drugiej metody w BASIC-u i wykonania prób. Od razu zdradzę tez swoje wyniki: posługiwałem sie interpreterem BASIC-a na IBM PC, przy jednokrotnym wykonaniu obliczeń dla wielomianu stopnia 100 różnica była niewidoczna gołym okiem Przy wykonaniu obliczeń 100 razy, bez użycia sekundnika widać, ze drugi z badanych programów jest szybszy. Bardzo się ucieszyłem tym rezultatem, gdyż jest on zgodny z

oczekiwaniami. Drugi program powinien być szybszy. Skoro tak, to spróbujmy zastanowić się, gdzie tkwi źródło tej większej szybkości. Odpowiedź jest prosta: zmieniliśmy algorytm. Krótko mówiąc, wróciliśmy do wyniku naszych rozważań z przed kilku miesięcy (Bajtek 4 i 5/87) — podstawowym źródłem efektywności programu jest dobry algorytm. Jednak aby móc mówić o dobrych czy złych algorytmach musimy mieć możliwość ich porównywania.

Powszechnie przyjętym kryterium jest ilość podstawowych operacji, które algorytm wykonuje aby znaleźć rozwiązanie. W przypadku obliczania wartości funkcji tymi podstawowymi operacjami będą oczywiście działania arytmetyczne.

Pierwszy z naszych algorytmów wymaga dla obliczenia wartości wielomianu stopnia n, wykonania n+1 potęgowań, n+1 mnożeń i n+1 dodawań. Drugiemu wystarczy 2n mnożeń i n dodawań. Zysk wynika ze znacznej różnicy czasu wykonania potęgowania i mnożenia (oczywiście na korzyść mnożenia), i wynosi:

n*(czas potęgowania-czas mnożenia)

Chociaż różnica może wydawać się nieco teoretyczna, to jednak wcale taka nie jest. Wielomiany tak często występują w praktycznych zastosowaniach, że nawet niewielki wzrost szybkości obliczeń daje efekt praktyczny. Efekt ten nie jest widoczny przy jednorazowym wykonaniu zadania, ale staje się bardzo znaczny jeśli wartość wielomianu trzeba obliczyć wiele tysięcy razy w jednym programie. Uzasadnionym staje się więc pytanie czy nie można by liczyć jeszcze szybciej?

Można, a algorytm który to umożliwia stanie się oczywisty, gdy zapiszemy wielomian w inny sposób. Zacznijmy od przykładu dla n=3:

$a_3x^3+a_2x^2+a_1x+a_0=((a_3x+a_2)x+a_1)x+a_0$

Proszę wymnożyć prawą stronę i przekonać się, że równość rzeczywiście zachodzi. Oczywiście w ten sam sposób można zapisać wzór ogólny:

$(((...(a_nx+a_{n-1})x+a_{n-2})x+...+a_2)x+a_1)x+a_0$

Nie będę przeprowadzał dowodu poprawności takiego zapisu, gdyż jest on powszechnie znany matematykom pod nazwą schematu Hornera (od nazwiska matematyka, który opublikował go w 1819 r.).

Przejdźmy od razu do zastosowania naszej nowej wiedzy w praktyce. Tym razem postąpimy jak rasowi projektanci i zamiast zaczynać od programowania spróbujemy policzyć ile operacji musi wykonać algorytm liczący wg tego schematu aby otrzymać wynik. Rachunek jest bardzo prosty, wystarczy się uważnie przyjrzeć wzorom. Mamy do wykonania n mnożeń przez wartość zmiennej x oraz n dodawań — spróbujcie na początek prześledzić to na podanym przykładzie, dla n=3**).

N mnożeń i n dodawań to znacznie lepiej niż poprzednio. Najlepiej jak można to zrobić. Jak się bowiem okazuje udało się udowodnić, że nie można obliczyć wartości wielomianu w jednym punkcie używając mniejszej liczby operacji arytmetycznych. Możemy więc nie łamać sobie dłużej głowy i ze spokojnym sumieniem napisać ostateczną wersję programu obliczającego wartość wielomianu.

Proponuję jednak odłożyć to na następny miesiąc, bo dziś nie sam program był najważniejszy, lecz praktyczne pokazanie, że o jakości programów nie decyduje sama biegłość w pisaniu instrukcji języka programowania, lecz również, (a może nawet przede wszystkim) umiejętność wymyślenia lub dobrania najlepszego algorytmu. Ważne też było pokazanie ciekawego algorytmu. Zabrakło natomiast miejsca na rozwinięcie kilku godnych uwagi szczegółów technicznych, dotyczących programowania, które pojawiły się po drodze, dlatego za miesiąc wrócimy do programowania algorytmu Hornera.

Andrzej Pilaszek

---

*) **Może się np. zdarzyć, że dla pewnych danych jeden z dwu programów jest szybszy, a dla innych danych szybszy będzie drugi.**

) **Oczywiście w poważnej pracy potrzebny byłby tu jakiś dowód, że tak jest naprawdę dla każdego n. Takie dowody istnieją, nie będę ich cytował, aby nie zmieniać tego artykułu w teoretyczną rozprawę.



WSZYSTKO O SCALAKACH

Włączając ulubione Spectrum czy Atari z pewnością zastanawialiście się nie raz, jak powstaje tak skomplikowane urządzenie, co kryje się pod jego obudową i jak ono w ogóle funkcjonuje. Tym razem zaprezentujemy kilka książek poświęconych układom scalonym, czyli elementom, bez których nie byłoby możliwe zbudowanie mikrokomputera.

Pierwsza z nich, to **„Układy scalone w pytaniach i odpowiedziach"** R. Ćwirko, M. Ruska i W. Marciniaka wydane przez WNT. Znajdziemy w niej bogate informacje na temat stosowanych w elektronice procesów wytwarzania oraz różnych typów układów scalonych i sposobów ich wykorzystania. Czytelnik ma okazję zapoznać się z nowoczesnymi układami scalonymi produkowanymi w technologii MOS, PMOS, NMOS lub CMOS, rozszyfrować znaczenie terminów: układ scalony bipolarny USB czy też układy hybrydowe.

Czytelników „Bajtka" na pewno szczególnie zainteresują układy cyfrowe, które są cegiełkami służącymi do budowy mikrokomputera. W dalszej części książki autorzy zajmują się właśnie podstawami algebry Boole'a i układami cyfrowymi TTL i TTLS, jak bramki, przerzutniki, liczniki czy rejestry oraz pamięciami półprzewodnikowymi RAM, PROM i EPROM.

Bliższe informacje o mikroprocesorach można znaleźć w książce **„Mikroprocesor w pytaniach i odpowiedziach"** K. Sachy i A. Rydzewskiego, którą szczegółowo przedstawiliśmy w numerze 8/87. Obie książki zredagowane są w bardzo przystępnej formie pytań i odpowiedzi, ilustrowane licznymi rysunkami, schematami i wykresami.

Trzecią pozycją, którą warto przeczytać, jest obszerne opracowanie Antoniego Niederlińskiego **„Mikroprocesory, mikrokomputery, mikrosystemy"** wydane przez Wydawnictwa Szkolne i Pedagogiczne. Książka ta zawiera dużo wiadomości o mikrosystemach i może okazać się przydatna różnym grupom odbiorców. Szczególnie dokładnie potraktowano w niej zagadnienie mikro — i makroarchitektury systemu mikrokomputerowego. Korzystanie z książki i zrozumienie trudnej tematyki jest możliwe nawet dla czytelnika mającego niewielką wiedzę o elektronice. Pomagają w tym m.in. starannie opracowane ilustracje.

Autor przyjął zasadę omawiania od podstaw pojęć z dziedziny informacji cyfrowej. Po przebrnięciu przez rozdziały ukazujące zasadę funkcjonowania rejestru, przerzutnika czy licznika mamy już odpowiednie przygotowanie, by zająć się architekturą mikroprocesora i mikrokomputera lub budową i działaniem układów wejścia — wyjścia. W drugiej części autor proponuje czytelnikowi rozpoczęcie samodzielnego programowania. Wstępem do poznania reguł tej trudnej sztuki są ciekawe rozdziały o strukturze i konstrukcji algorytmu oraz o asemblerze mikroprocesora 8880 i Basicu. Bardzo cenna wiadomość znajduje się w części opisującej sposób wymiany informacji między elementami mikrosystemu. Omawiane tu różne protokoły komunikacyjne będą użyteczne dla tych, którzy spróbują podłączyć nietypowe urządzenie zewnętrzne do swojego komputera.



Projektantom oraz studentom informatyki polecilibyśmy wydaną przez WNT książkę niemieckiego autora Rudolfa Hedtke **„Systemy mikroprocesorowe"**. Książka zajmuje się niezawodnością sprzętu, sposobami określania i przewidywania na etapie projektowania intensywności uszkodzeń elementów mikrokomputera. Pokazane są w niej także metody testowania pojedynczych podzespołów, a także całego systemu oraz urządzeń peryferyjnych i pamięci.

Duże zainteresowanie i wiele pytań o sposób programowania w asemblerze świadczy o ogromnej popularności tego języka. Zdobycie literatury dotyczącej mikroprocesora Z-80 nie stanowi w chwili obecnej większego problemu. Listę rozkazów i podstawowe wiadomości o posługiwaniu się językiem maszynowym można znaleźć w instrukcjach obsługi Spectrum, zeszytach wydawanych przez Stronnictwo Demokratyczne, książce „Mikroprocesor w pytaniach i odpowiedziach", skrypcie SOETO „Oprogramowanie podstawowe ZX-Spectrum — DEVPAC 3 MONS3)" lub omówionej w poprzednim numerze Bajtka książce „Mikroprocesor Z-80". Dziwi natomiast zaniedbanie licznego grona użytkowników komputerów z rodziny Commodore, Apple, Laser, Acorn czy Atari. Listę rozkazów bardzo rozpowszechnionego mikroprocesora 6502 znajdziemy tylko w „Atari BASICU" W. Miguta lub nielicznych artykułach np. w „Komputerze". Dużego zapotrzebowania na literaturę o 6502 na pewno nie zaspokoi książka Jana Ruszczyca **„Asembler 6502"**, wydana w niewielkim nakładzie 5000 egz. przez SOETO. Jest to dobrze napisany podręcznik programowania w asemblerze, zawierający obok opisu rozkazów i trybów adresowania wiele przykładów procedur i ćwiczenia do samodzielnego rozwiązania. Technika programowania została omówiona w łatwy do zrozumienia sposób. Podprogramy skonfrontowano z odpowiadającymi im strukturami Basica, co bardzo pomaga w ich analizie. Ze względu na dużą popularność Atari dołączono także rozdział poświęcony układom ANTIC i GTIA, a także wiele uwag, sposobie tworzenia obrazu, wykorzystania grafiki graczy i pocisków oraz o przerwaniach. Książka ta uczy racjonalnego posługiwania się asemblerem, który jest językiem pozwalającym „lepiej zrozumieć architekturę mikrokomputera i wewnętrzne mechanizmy działania" oraz „głębiej wniknąć w strukturę języków wysokiego poziomu i efektywnie w nich programować".

(j.j.).

Robert Ćwirko, Mirosław Rusek, Wiesław Marciniak — „Układy scalone w pytaniach i odpowiedziach", WNT, Warszawa 1987. Wyd. II. Nakład 60000 egz. Cena 590 zł.

Antoni Niederliński — „Mikroprocesory, mikrokomputery, mikrosystemy" — Wydawnictwa Szkolne i Pedagogiczne, Warszawa 1987, Wyd. I. Nakład 10000 egz. Cena 657 zł.

Rolf Hedtke — „Systemy mikroprocesorowe" — przełożył Andrzej Dworak. WNT, Warszawa 1987, Wyd. I. Nakład 70000 egz. Cena 500 zł.

Jan Ruszczyc — „Asembler 6502" — SOETO, Warszawa 1987, Wyd. I. Nakład 5000 egz. Cena 910 zł.



SŁOŃ I SPRYTNA MYSZKA

Walka o mózgi trwa. Pochodzący z Tajwanu Steve S. Chen opuścił największą na świecie firmę superkomputerową Craya, by wkrótce dostać się pod skrzydła głównego giganta od produkcji masowej — IBM.

„Big Blue" gotowy jest zainwestować miliony, by stworzyć za kilka lat bardziej powszechny, komercyjny superkomputer najwyższej generacji, oparty na wielu równoległych procesorach i wykorzystujący połączenia świetlne. Czy niezwykły talent Chena, przy wsparciu pieniędzy i ekspertów IBM, rozwikła zagadkę i to w ciągu 5 lat? — zastanawia się tygodnik DER SPIEGIEL.

Na każdym z zebraniu akcjonariuszy spółki Cray Research stawiano jest pytanie, co stanie się z firmą jeśli jej główny udziałowiec Seymour Cray dostanie się np. pod koła ciężarówki. 61-letni twórca najszybszego komputera na świecie opowiada zwykle, że również jego przeraża taka perspektywa.

Przez kilka ostatnich lat istniało zabezpieczenie dla „ojca chrzestnego" branży superkomputerowej i jego firmy w Minneapolis, w amerykańskim stanie Minnesota. Od 1979 roku poza Cray'em również Steve S. Chen, „syn chrzestny" w hierarchii elektronicznych czarodziejów, tworzył dzieła sztuki z chipów, tchnąc w nie życie. Jednakże we wrześniu ubiegłego roku geniusz wyrwał się z zaprzęgu. Zawiedziony polityką firmy, Chen pokazał plecy przedsiębiorstwu Cray Research. [illegible] twórcy niezwykłych konstrukcji [illegible] wybitnie [illegible] superkomputery. Akcje [illegible] spadły. [illegible]

Do końca marca [illegible] nowe z Chenem [illegible] stałe [illegible] ce z centrali IBM w Armonk, w stanie Nowy Jork, mówią, że ten największy na świecie producent komputerów wchodzi do powstającej właśnie spółki Chena „Supercomputer System Inc." w Fauclaire, w amerykańskim stanie Wisconsin. W zamian za pakiet udziałów o nieznanych jeszcze rozmiarach IBM chce sobie zapewnić wgląd w tajemnice kuchni superkomputerowych magów. Doborowa grupa ekspertów ze stajni IBM ma — zgodnie z zawieranym porozumieniem — współpracować z pochodzącym z Tajwanu twórcą superkomputerów.

W ciągu 5 lat ze związku między potężnym, opanowującym rynki, słoniem a sprytną myszką winien narodzić się pierwszy, w pełni dojrzały produkt nowej serii — komputer najwyższej generacji, „pożeracz liczb", o szybkości kilkuset razy większej niż obecny rekordzista świata „Numbercruncher" z wytwórni Cray'a, o nazwie Cray 2.

Oto recepta Chena na planowany superkomputer: 64 wysoko wydajne procesory winny wspólnie i równocześnie zajmować się programem, ultraszybkie połączenia świetlne mają zastąpić klasyczne przewodowe połączenia.

Maszynę działającą zgodnie z tymi zasadami chciał Chen opracować również dla byłego pracodawcy — Cray'a, któremu jednakże koncepcja ta wydała się tak ryzykowna (koszty badań szacowano na 100 milionów dolarów), że wstrzymał projekt, określany w skrócie MP („Multiple Processor"). Dla Chena było to wystarczającym powodem, by opuścić Cray'a.

Fakt, iż konserwatywny kolos, jakim jest IBM, zdecydował się na zakup talentu Chena, zaskoczył ekspertów. Wraz z tym czołowym konstruktorem, IBM decyduje się na [illegible] równocześnie wielkie skoki w nieznane — na przejście w wąski krąg twórców maszyn typu „Numbercruncher" i na wkroczenie w ryzykowną dziedzinę komputerowej [illegible]

[illegible] 197[illegible] Seymour Cray ustanowił po raz pierwszy standard w dziedzinie superkomputerów w postaci maszyny obliczeniowej Cray 1, IBM był na tym rynku, wyznaczanym odtąd przez Craya, jedynie stojącym na uboczu obserwatorem. Jeszcze w 1987 roku, gdy IBM zaprezentował własną maszynę o najwyższej wydajności, a mianowicie model „3090—600 FVF" koncern podtrzymywał podstawowe twierdzenie, że komputer o strukturze równoległej dla celów komercyjnych byłby zbyt kosztowną i zbyt ryzykowną techniczne przygodą.

IBM musiała więc z pewnością przetrawić na nowo całą sprawę, aby przywrócić do łask postulat schematu równoległego — stwierdza Denny Hillis, enfant terrible wśród konstruktorów superkomputerów. Decyzja „Big Blue" w sprawie Chena oznacza zdaniem Hillisa olbrzymi krok naprzód w zakresie badań nad strukturą równoległą" — *wszyscy muszą teraz bardzo przyspieszyć, gnani strachem przed IBM.*

Dotychczas obowiązująca zasada matematyczna, wynaleziona w latach czterdziestych przez amerykańskiego matematyka Johna von Neumanna, zgodnie z którą jeden centralny procesor opracowuje kolejno, krok po kroku, informacje zawarte w pamięci roboczej — wyczerpała możliwości twórcze. Seymour Cray i jego prawa ręka Chen osiągnęli w swych arcydziełach X-MP i Cray-2 granice zawartych w niej możliwości.

W ostatnim dziele Chena, nazwanym Cray X-MP, już cztery procesory zajmują się prowadzeniem obliczeń. Na tyle, na ile pozwala program, procesory te dzielą między siebie czynności logiczne i dzięki temu osiągają rekord w wysokości ponad miliarda operacji zmiennoprzecinkowych na sekundę. Maszyny o takiej wydajności oferuje jak do tej pory jedynie pięciu producentów — poza Crayem występują na rynku jedynie firma Control Data i trzy firmy japońskie.

Dzięki technice czteroprocesorowej firma Cray, która dotąd pokrywała dwie trzecie zapotrzebowania na super komputery na rynku światowym, wkroczyła sama na pierwszy stopień „architektury równoległej". Bardziej niż ostrożne kroki prowadzące stamtąd do poziomu 8, a następnie 16 procesorów wydały się jednak Cray'owi i tak na razie zbyt ryzykowne.

Na — jak dotąd — najbardziej radykalny odwrót od zasady klasycznej odważył się przed niespełna dwoma laty, mający wówczas 30 lat, konstruktor komputerów Hillis. W 1983 roku założył on w Canbridge w stanie Massachucsetts firmę „Thinkung Machine Corp", a następnie w 1986 roku przedstawił własny komputer równoległy „Connection Machine". Na czarnej, matowej kostce obliczeniowej 65536 procesorów przetrawiało w nim wspólnie pokrojony na kawałeczki program.

Za około jednej czwartej ceny komputerów Cray'a, z których każdy kosztuje dobre 12 milionów dolarów, otrzymuje się dzięki temu urządzenie o porównywanie wysokich osiągach. Jak dotąd system podziału pracy między procesorami nadaje się oczywiście jedynie do niewielu programów specjalnych.

Chen natomiast chce wynaleźć opartą na systemie równoczesnym super-maszynę obliczeniową „na codzienny użytek", nie zaś — maszynę wyścigową do pokonywania odcinków próbnych. Tego rodzaju zadanie wymaga jednak nie tylko ducha pionierskiego, ale i dobrych pieców, takich, jakie może zapewnić koncern IBM.

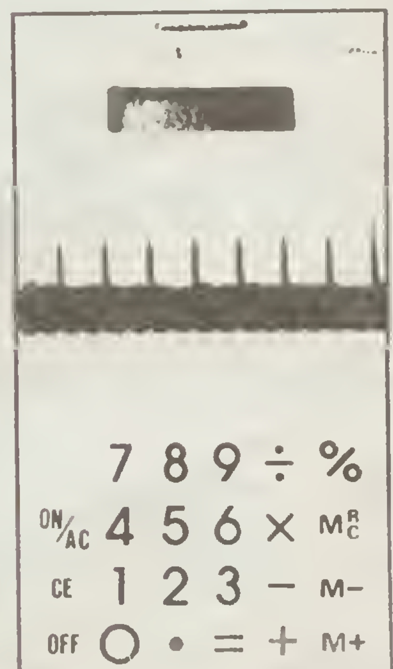
Dlaczego niektóre problemy rozwiązywane są przez „komputery paralelne" setki razy szybciej, inne zaś wymagają tyle samo czasu, co maszyny zbudowane zgodnie z zasadą von Neumanna? — Tak oto ekspert z IBM Abraham Poled sformułował jedną z zagadek systemu równoległego. Jeszcze inny dylemat, odnoszący się do programowania, brzmi: *co należy czynić w przypadku komputerów systemu równoległego, jeśli wiele procesorów wywoła w tym samym czasie te same informacje z pamięci?*

Autorów tego rodzaju wątpliwości ostro gani Hillis, który mówi o „filozoficznym bajdurzeniu" ludzi bez odwagi. Wraz z wejściem Chena — ma nadzieję ten szyderca — całe gadanie się zakończy: węzły problemów przetnie się najlepiej demonstrując kilka logicznych dowodów.

opr. (f)

Te zapałki można bez obawy dać do zabawy nawet małemu dziecku. Każdy, kto przy ich pomocy spróbuje rozniecić ogień, odkryje ze zdziwieniem, że w pudełeczku ukryto kalkulatorek. Być może przemyślny i dowcipny konstruktor w ten właśnie sposób pragnął uchronić nas przed skutkami beztroskich igraszek pozostawionych w domu bez opieki maluchów. Poza tym zawsze warto mieć pod ręką proste urządzenie, które realizuje cztery podstawowe działania.

(j.j)



JAK TO ROBIĄ INNI



WYCZYTANE Z RĘKI

Kiedy 8 lat temu odebrałem pierwszy całkowicie zredagowany i napisany przy pomocy osobistego komputera list mego amerykańskiego przyjaciela, miałem wrażenie obcowania z czymś nadzwyczajnym, wręcz sensacyjnym. Dziś nawet w Polsce specjalnie nie szokuje arkusz papieru zapełniony symbolami, znakami, czy rysunkami wypełzającymi z urządzenia, które zwykliśmy nazywać drukarką. Ale żeby były na nim kopie fotografii?

W 1984 roku pojawiły się w Stanach Zjednoczonych powszechnie dostępne oprogramowania umożliwiające manipulację tekstem i włączanie w jego ustępy zreprodukowanych zdjęć, szkiców, czy nawet pozostawionych w pamięci systemu obrazów. Pozostawionych dzięki współpracy z przyłączonymi do mikrokomputera specjalnymi scannerami przetwarzającymi całe stronice książek. Dzięki temu wszystkiemu można się było stać właścicielem domowego kserografu i to tak doskonałego, iż pozwalającego na dowolną obróbkę wprowadzonego do pamięci obrazu. Kserografu tańszego od powszechnie stosowanych olbrzymów, obdarzonego na domiar wszystkiego możliwością przesyłania na odległość, dzięki połączeniu komputera z siecią telefoniczną, tego, co leżało pod scannerem.

Właśnie w ten sposób w Stanach Zjednoczonych rozpoczął się, obecnie rozpowszechniony nie tylko w Ameryce, ruch edytorów domowych pisemek komputerowych. Gazetek przesyłanych telefonicznymi łączami znajomym i przyjaciołom. Mikrokomputery zrewolucjonizowały także funkcjonujące do tej pory dzięki kserografom miesięczniki hobbystyczne wydawane przez miłośników kotów, psów starej broni, historii lotnictwa, antycznych samochodów, etc..., określane w Stanach Zjednoczonych jako „non profit publications", czyli publikacje nie obliczone na zysk i utrzymujące się ze składek członkowskich czytelników. Pisma zapaleńców przechodziły więc stopniową elektroniczną metamorfozę. Aby nie być gołosłownym powiem, że śledziłem ją na przykładzie wydawanego od lat siedemdziesiątych w Kalifornii „Small Air Forces Observera" — czyli „Obserwatora Sił Lotniczych Małych Państw" — periodyku historyków-amatorów dziejów wojskowej awiacji wydawanego zrazu kserograficznie przez Jima Sandersa.

Ze zdjęciami i rysunkami zamieszczanymi w książkach był jednak zawsze podstawowy problem — najlepiej pod scanner wchodziły pojedyncze, płaskie kartki. **Ideałem byłoby wyrwanie ich choćby na chwilę z kopiowanego wydawnictwa, a potem powtórne wklejenie do całej reszty dzieła. Niestety, z oczywistych względów, takie operacje nie były możliwe w przypadku wielu cennych pożyczonych, czy też własnych książek. Aby udowodnić czytelnikom na czym polegała cała trudność powiem co się stanie, gdy spróbujemy dokonać kserograficznej odbitki strony cienkiej, klejonej grzbietowo broszury. Jeżeli rozłożymy ją całkowicie i spłaszczymy, pęknie spoiwo i strony rozsypią się niczym talia kart. Oberwiemy z całą pewnością od właściciela sprofanowanego w ten sposób wydawnictwa.**

Cóż więc było robić? Należało skonstruować skanner nie niszczący delikatnych książek, a przenoszący do pamięci komputera zawarte w nich obrazy. **Takie urządzenie pojawiło się właśnie na rynku amerykańskim. Składa się z ręcznego czytnika obrazów opracowanego przez Japan Mitsumi Corporation, połączonego z interfejsem dającym się podłączyć do dowolnego komputera rodziny IBM, lub kompatybilnego z IBM, a produkowanym przez Diamond Flower Electric Instrument Co. z Sacramento w Kalifornii.** Ta druga firma rozprowadza po terenie Stanów Zjednoczonych ów Handy Scanner HS-1000 licząc sobie w detalu 299 dolarów za cała instalację.

Jak działa HS-1000? Plastykowe pudełeczko mieszczące się w dłoni, przypominające mikrofon polowej radiostacji „czyta" podstawiane pod jego okienko szkice, fotografie, rysunki, fragmenty planów z powierzchni czasopism, czy książek i przesyła zapamiętany obraz do komputera. Ten wyświetla to, co odebrał na ekranie, a więc można powtórzyć zabieg zapamiętywania, jeśli nie jesteśmy zadowoleni z jego rezultatów. Dalej możemy z naszym obrazem robić wszystko na co pozwala edytorski program komputera — zmniejszać, rozciągać, wkładać z boków ścieśnianego tekstu. W miarę potrzeb i fantazji.

Przysposabiając HS-1000 do operacji „czytania" nastawić musimy żądany kontrast obrazu, a następnie przejeżdżamy naszym ręcznym scannerem po obrazie, który chcemy przenieść do komputera. Zapamiętana zostanie ścieżka obrazu o wymiarach 2,5 cala (63,5 mm) na 10 cali (254 mm). Jeżeli uważamy, iż to za mało, przesuwamy urządzenie nad pozostałą nie wczytaną jeszcze częścią obrazu, robiąc to tak długo, aż całość znajdzie się na ekranie monitora. I teraz możemy całkowicie pewni swego, zacząć zabawę w domowego wydawcę.

Tyle przyjemności dostarcza już teraz amerykańskim komputerowym fanom i specjalistom spółka Diamond Flowers. **Zapowiada wszakże coś jeszcze doskonalszego już w tym roku. Podobno rozpoczęła badania systemu podręcznego scannera umożliwiającego optyczne „wczytywanie" wydrukowanego tekstu. Rynkowe wydanie testowanego prototypu ma zostać rzucone do sklepów jeszcze w 1988 roku. O cenie nie ma na razie mowy.**

Wojciech Łuczak

PRAVEC

Seria bułgarskich komputerów „Pravec" wzięła swą nazwę od miasta położonego 70 km od Sofii, gdzie znajduje się centrum ich produkcji. Prócz 15 nowoczesnych zakładów powstała szkoła zawodowa, jeden z wydziałów politechniki oraz instytut cybernetyki.

Produkcja PC „Pravec", którą rozpoczęto w 1981 roku od ośmiobitowych modeli „domowych" 8D, profesjonalnych 8M, edukacyjnych 8A [illegible] — szesnastobitowych „Pravec 16" [illegible]ągnąć ma w najbl[illegible] n poziom 100 tys sztuk rocznie

— Po dwu latach produkcji komputerów 16 bitowych trudno mówić o jakimś szczególnym doświadczaniu, ale jeśli chodzi o niezawodność nasze PC mieszczą się w światowym standardzie — twierdzi inż. **BORISLAW KUJUMDŻIJEW** promujący z ramienia centrali handlu zagranicznego „Isotimpex" bułgarskie komputery na rynkach zagranicznych

— Patrząc na to z handlowego punktu widzenia — kontynuuje jego kolega z centrali inż **SVETLOZAR GRIGOROW — można powiedzieć, że nasi klienci nie narzekają na jakość zakupionego u nas sprzętu. Znaczy to bardzo wiele, bo przecież w RWPG jesteśmy pionierami w tej dziedzinie i nikt nie oczekiwał, że wszystko pójdzie jak z płatka.**

— Dochodząc do produkcji seryjnej jednostek 16 bitowych — wspomina inż. Kujumdżijew — **dokonaliśmy ogromnego skoku naprzód. Nie stało się tak w ciągu roku czy dwóch. Bułgaria koncentrowała się na elektronice od lat i, co jest szczególnie ważne — robiła to celowo i systematycznie. Te doświadczenia stworzyły solidne podstawy naszych obecnych osiągnięć i spowodowały, że rezultaty nie mogły być inne. Dotyczy to w tym samym stopniu mikro- jak i minikomputerów oraz dużych jednostek wraz z urządzeniami im towarzyszącymi.**

Drugim ośrodkiem produkcji mikrokomputerów są zakłady STO w Wilke Trnovo. Produkowany tam sprzęt to profesjonalne 16 bitowe EC1831 i 1832. W ubiegłym roku zakłady te osiągnęły już planowaną zdolność produkcyjną — 1000 sztuk kompletnych systemów.

PRAVEC 8

Komputery tej serii opierają się na 8-bitowym procesorze Motorola 6502. Najprostsza wersja — 8D — przeznaczona jest do nauczania podstawowego (programowania). Jako monitor może służyć ekran telewizora (wystarczając rozdzielczość 240 na 200 pkt.); pamięć zewnętrzna — kaseta

Bardziej zaawansowany typ 8M, zgodny z Apple II jest modelem szeroko rozpowszechnionym w bułgarskich szkołach. Ponieważ produkcję tych modeli rozpoczęto w 1981 roku, oprogramowanie dostępne jest w dużym wyborze. W przypadku tego mikrokomputera i jego wersji edukacyjnej — 8A dostarcza się z zasady kompletny system z monitorem monochromatycznym drukarką M80 oraz jedną stacją dysków

Dane techniczne:

— mikroprocesor	Motorola 8502
— pamięć ROM	12 KB
— pamięć RAM	64 KB
— pojemność (modele 8M i 8A)	140 KB
— tekst	24 wiersze po 40 lub 80 znaków
— grafika	280 × 192 pkt/cal, 8 barw
— system operacyjny	DOS 3.3, CP/M, UCSD-p
— język	Basic, Assembler i makroassembler, Pascal, LOGO, Fortran, Cobol, Pilot.

PRAVEC 16

Występuje w dwu odmianach PRAVEC 16N z monitorem o rozdzielczości 720 na 348 pkt /cal i przenośny PRAVEC 16S z monitorem o przekątnej 23 cm, który stanowi integralną część komputera.

PRAVEC 16N występuje w czterech wersjach:

a — jako terminal

b — z 1 stacją dysków o poj. dyskietki 320 do 360 KB

c — z 2 stacjami dysków

d — z jedną stacją dysków elastycznych i dyskiem twardym 10 MB

Dane techniczne:

— Mikroprocesor	Intel 8088
— Koprocesor	Intel 8087
— Zegar	4,77 MHz
— RAM	256—640 KB
— ROM	40KB
— Tekst	25 wierszy 40 lub 80 znaków
— System operacyjny	PC-DOS, MS-DOS, CP//M-86, Concurrent, CP/M, USCD-p, UNIX apod.
— Języki	Basic, Pascal, Cobol, Fortran.

PC EC 1831 i EC 1832

Są najlepszymi wśród 16 bitowych produkowanych w Bułgarii. Zgodne z PC XT oba typy różnią się tym, że w modelu EC 1832 jedną ze stacji dysków zastępuje dysk twardy

Dane techniczne:

— Mikroprocesor	Intel 8088
— Koprocesor	Intel 8087
— Zegar	4.77 MHz
— RAM	128 do 640 KB
— ROM	64 KB
— Dysk twardy	ES 5300/5508 — 10 MB
— Dysk elastyczny	ES 5088/5321
— Tekst	25 rzędów; 40—80 znaków
— Grafika	640 × 200 pkt /cal monochr 320 × 200 pkt./cal barwny
— System operacyjny	DOS-PK, MIKROS 86, DOS-P
— Język programowania	Basic, Assembler, Fortran, Pascal, Cobol, „C"
— Interface	RS 232C (C2)

opracował Pen.

WSZYSTKO DLA WSZYSTKICH

Dziękuję TV — Video, Kserokopiarki, Microcomputery Berlin Zachodni 30, Wurzburgerstr. 2, tel. 214-11-58 za dostarczone kalkulatory, układy scalone. Bew.

K-45

LEGITYMACJA NIEKONIECZNA

Rozmowa z Krzysztofem Krupą, sekretarzem Rady Ogólnopolskiej Federacji Klubów Komputerowych Młodych Mistrzów Techniki.



— W zeszłym roku „Złotą Dyskietkę BAJTKA" zdobył klub MERIZAP z Ostrowa Wielkopolskiego zrzeszony w Federacji. Czy i w tym roku przewidujesz podobny sukces?

— Trudno mi przewidywać decyzję jury. Wiem jednak, że nasze kluby mają w tego typu współzawodnictwach wiele do pokazania. I niejeden jest tak dobry jak MERIZAP.

— Działacie w ramach ZSMP. Czy tworzenie klubów komputerowych nie wygląda przypadkiem na próbę przekupienia młodych ludzi bardzo atrakcyjnymi zabawkami?

— Z pewnością nie o to nam chodzi. Człowiek zainteresowany informatyką trafia przecież do klubu komputerowego, a nie na zebranie koła i dopuszczenie go do sprzętu nie zależy od legitymacji naszego Związku. Poza tym, do Federacji należy wiele klubów niezwiązanych z Organizacją. Utworzono ją przy ZSMP, ponieważ zgodnie z Ustawą o wynalazczości, właśnie w ramach Związku Socjalistycznej Młodzieży Polskiej, działa Turniej Młodych Mistrzów Techniki, powołany do wspierania innowacyjnych, a więc i informatycznych pasji młodzieży.

— Odwiedzając kluby komputerowe w całej Polse zauważyłem, że każdy z nich jest nieco inny, każdy pracuje w innych warunkach, jaki jest więc sens łączenia ich działań?

— Kluby różnią się wprawdzie między sobą, problemy mają jednakże bardzo podobne. Brakuje wydawnictw książkowych z informatyki, szczególnie dobrych i tanich podręczników. Ceny sprzętu, zwłaszcza klasy PC, wahają się nieustannie. Łatwo też kupić złom zamiast komputera czy dyskietek. Klubowiczom, tym bardziej ambitnym, brakuje dostępu do szkoleń na odpowiednim poziomie, kontaktów z najnowszymi programami itp. Dotyczy to w pierwszym rzędzie członków klubów działających poza ośrodkami akademickimi. Nasza Federacja stara się trudności te rozwiązywać. Likwidacja ich siłami pojedyńczego klubu nie jest możliwa.

— Na co mogą z Waszej strony liczyć kluby?

— Na pomoc i zasoby Centralnego i Wojewódzkich Klubów Komputerowych, o czym już mówiliśmy. Mam tutaj na myśli przede wszystkim oprogramowanie oraz wszelką pomoc organizacyjną, w tym również prawną. I jeszcze jedna uwaga — pytasz na co mogą liczyć kluby — co sugeruje instytucjonalny charakter Federacji, która przecież jest raczej masowym, społecznym ruchem. To istotna sprawa, ponieważ nasze możliwości w znacznym stopniu zależą od współdziałania i stopnia koordynacji wspólnych przedsięwzięć. Od jednomyślności w określaniu i artykułowaniu jednakich dla wszystkich interesów. Ta sprawa ma zasadnicze znaczenie obecnie, gdy głównym problemem ruchu nie jest dostęp do ZX 81, a tworzenie banków informacji i dostęp do nich, nie kurs BASIC'a na MERITUM I, a nauka d'BASE III+. Po prostu klub to dziś już nie tylko zabawa, a w coraz większym stopniu rzetelna nauka wykorzystania komputera w miejscu pracy czy w szkole, co wymaga odpowiedniej organizacji, dużych środków i pewnej planowości w działaniu. I na umacnianie tych tendencji w Federacji, można bezsprzecznie liczyć.

— Wydaliście już dwie pozycje w serii „Mikrokomputery", w ramach biblioteczki Powszechnego Ruchu Inicjatyw Innowacyjnych POLIN. Jakie macie zamierzenia na najbliższą przyszłość?

— Mamy przygotowane cztery następne pozycje dotyczące języków programowania, programów użytkowych i metodyki pracy klubu komputerowego. W sumie na ten rok 12 tytułów. Mamy autorów i drukarnię, brakuje nam tylko ... papieru.

— Czy każdy z klubów należących do Federacji otrzymuje te pozycje?

— Oczywiście, każdy z naszych klubów otrzymuje odpowiednią ich liczbę. Na razie nieodpłatnie.

Redagujemy też pierwszą w Polsce „gazetę komputerową", w postaci programu na ZX SPECTRUM. Jest ona emitowana przez Polskie Radio, w audycji RADIOKOMPUTER, raz w miesiącu.

— Jakie są Wasze plany na okres najbliższy i trochę dalszy?

— Najważniejsze dla przyszłości ruchu KK MMT jest posiedzenie Komitetu do Spraw Nauki i Postępu Technicznego, który w najbliższym czasie będzie rozpatrywał problemy wychowania innowacyjnego młodzieży, w tym również edukacji informatycznej prowadzonej przez Kluby Komputerowe Młodych Mistrzów Techniki. Jesteśmy w trakcie opracowywania materiału o naszych dokonaniach oraz potrzebach.

Poza tym wyposażamy kluby wojewódzkie. W ubiegłym roku zakupiliśmy 130 mikrokomputerów Amstrad. W roku bieżącym, nadal będziemy nabywać dla klubów Amstrad'y, jak również większe ilości IBM'ów. Chcemy w kwietniu uruchomić Centralny Klub Komputerowy. Będzie tam miejsce na wymianę oprogramowania, kompletowanie literatury. Chcemy stworzyć bibliotekę, gdzie każdy będzie mógł nie tylko wziąć do ręki wydawnictwa krajowe i zagraniczne, ale także zrobić kopię ich fragmentów, zdobyć informacje co warto kupić, gdzie, kiedy i za jaką cenę.

Złożyliśmy zamówienie na opracowanie taniego modemu galwanicznego, który chcemy wytwarzać z przeznaczeniem dla naszych klubów. Obecnie testujemy prototyp.

W marcu rozstrzygniemy drugą już edycję konkursu na najlepiej pracujący klub komputerowy. Nagrodą Główną jest tym razem IBM PC XT. To tyle jeżeli chodzi o najbliższą przyszłość.

— W ubiegłym roku pośredniczyliście w zakupach sprzętu dla klubów. Wiem, że był to sprzęt bardzo tani. Czy nadal macie takie możliwości?

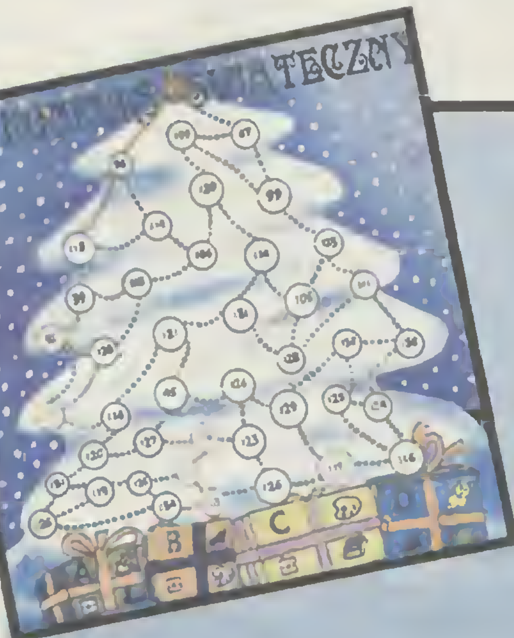
— Udało nam się wówczas wynegocjować duże obniżki cen i część naszych klubów zdążyła z tego skorzystać. Niestety, w listopadzie, w związku ze zmianami na rynku, ceny sprzętu znacznie się podniosły. Sądzę jednak, że w niedługim czasie dojdziemy do porozumienia z jakąś firmą handlową i znów będziemy w stanie przedstawić naszym klubom atrakcyjną ofertę zakupów. Na razie negocjujemy.

— Kluby Komputerowe Młodych Mistrzów Techniki, znajdują się w całej Polsce. Gdzie młody człowiek, zainteresowany informatyką znaleźć może informacje o Waszych klubach, w jego mieście czy gminie?

— Już niedługo — jak sam wiesz — rozpoczniemy ich prezentację na łamach BAJTKA. Niezależnie od tego, informacji takiej udzielają Wojewódzkie Biura TMMT, które mieszczą się przy Zarządach Wojewódzkich ZSMP. Wystarczy więc wziąć do ręki książkę telefoniczną. Służę też swoim wykazem adresów pod telefonem 265401 wew. 362 w Warszawie.

Rozmawiał:
Roman Poznański

Spieszymy poinformować Czytelników, że sklep „BAJTKA" w Bytomiu zmienił swoją siedzibę. Obecnie mieści się przy ul. Koniewa 6. Tel. 81-57-01.



KONKURS ŚWIĄTECZNY ROZSTRZYGNIĘTY!

Już od dawna listonosz doręczający korespondencję do redakcji „Sztandaru Młodych" nie miał tyle pracy. Na ogłoszony przez nas w „Bajtku" z grudnia ub. roku Konkurs Świąteczny wpłynęło ponad 15 tys. odpowiedzi. 23 lutego br. przed kamerami „Teleexpresu" odbyło się losowanie nagród. Oto pełna lista szczęśliwców.

Główną nagrodę komputer Atari 65XE z magnetofonem wylosowała Bożena Potoczny z Oławy.

Posiadaczami plecaków „Atari" stali się Wojciech Drwal z Tarnowa, Małgorzata Nowak i Jerzy Pieńkowski z Warszawy oraz Anna Strzyżewska z Góry Kalwarii.

Kasety z programami wylosowali Stanisław Nezveda z Pragi, Adam Lipszyc z Michalina, Tomasz Wilk z Katowic i Tomasz Oczkoś z Gorzowa Wielkopolskiego.

Książki o tematyce komputerowej przypadły: Sobiesławowi Gabarze z Bydgoszczy, Piotrowi Guzkowi z Chełmna, Piotrowi Grzelakowi z Łodzi, Krzysztofowi Bartzowi z Opola Lubelskiego, Jakubowi Osińskiemu z Warszawy, Arkadiuszowi Baronowi z Rudzińca, Waldkowi Kozłowskiemu z Elbląga, Krzysztofowi Wachali z Lwowka Śląskiego, Juliuszowi Zajdzie z Wrocławia, Piotrowi Kozłowskiemu z Białegostoku, Marlenie Władyczko z Gryfina, Mariuszowi Gruzie z Warszawy, Grzegorzowi Landsmanowi z Gdyni, Maciejowi Knulowi z Łaska, Pawłowi Witkowskiemu z Warszawy, Robertowi Woźniekowi z Lubuska, Ireneuszowi Rogale z Gdańska, Jarosławowi Charcholisowi z Bartoszyc, Wojciechowi Wiatrowskiemu z Nidzicy i Krzysztofowi Kamińskiemu z Wrocławia.

Ci, którzy nie wylosowali żadnej nagrody mogą zawdzięczać to bądź nieprawidłowemu rozwiązaniu (hasłem konkursu było „Bajtek Twoim przyjacielem") bądź... brakowi szczęścia. Może dopisze im ono przy rozwiązywaniu Konkursu Wielkanocnego.

Triumfatorom gratulujemy!



ROBOT EDUKACYJNY

MICROROBOT-1

Na przestrzeni ostatnich kilku lat obserwowany jest szybki rozwój małych robotów, zwanych robotami edukacyjnymi. Można wskazać dwie główne przyczyny tego rozwoju: — wzrost produkcji i zastosowań robotów przemysłowych, — burzliwy rozwój tanich komputerów osobistych. Roboty przemysłowe wraz z oprzyrządowaniem są urządzeniami drogimi. Jednocześnie rosnąca szybko liczba zastosowań tych robotów sprawia, że wzrasta zapotrzebowanie na małe i znacznie tańsze roboty treningowe o zbliżonych cechach funkcjonalnych. Roboty treningowe używane są w zakładach stosujących roboty do wstępnego przeszkolenia personelu, na uczelniach a także w szkołach. Urządzenia takie także stymulują studentów (uczniów) do prób z konstruowaniem innych, podobnych urządzeń. Od czasu, gdy mikrokomputer można wbudować do niewielkich zabawek znalazło się wielu hobbystów budujących własne kroczące maszyny, jeżdżące myszy, żółwie i inne inteligentne urządzenia. Tak narodziła się idea konkursów, łączących przyjemne z pożytecznym.

Konkursom tym patronują poważne instytucje, takie jak IEEE. Zadaniem uczestników turnieju MICROMOUSE jest skonstruowanie robota potrafiącego znaleźć najkrótszą drogę w labiryncie w ograniczonym czasie. Wymiary labiryntu są narzucone, w związku z tym mysz nie może być szersza i dłuższa niż 25 cm. Dodatkowym utrudnieniem jest wymagania całkowitej niezależności. Każdy robot musi mieć własne zasilanie i napęd i musi być samosterowany. Turnieje tego typu rozgrywane są od 1980 r. i obecnie najlepsze myszy dysponują czujnikami podczerwieni, radarem ultradźwiękowym, możliwością przyśpieszeń na długich prostych (osiągają prędkość do 3 m/sek.) i hamowania przed zakrętem. W tym turnieju byłoby ciężko wygrać, ale można spróbować w micro ping-ponga. Na razie wygrywały roboty, którym w ogóle udało się trafić w piłeczkę. Wszystkim zainteresowanym czytelnikom, chcącym brać udział w takich konkursach polecam szczegółową lekturę INFORMATYKI 1/86.

Ale powróćmy do robotów edukacyjnych. Konfiguracja kinematyczna robota edukacyjnego jest mniej lub bardziej zbliżona do konfiguracji ludzkiej ręki. Typowy manipulator posiada zwykle 5 stopni swobody, co zapewnia mu podobieństwo do robota przemysłowego. Główne różnice występują przy porównywaniu wymiarów (kilkadziesiąt centymetrów — 1.5-2 m), a co za tym idzie przestrzeni roboczej, udźwigów (poniżej 1 kg — do 20 kg), dokładności i oczywiście ceny (kilkaset, kilka tysięcy dolarów — kilkadziesiąt tysięcy dolarów). Podstawowym wymogiem przy budowie robota edukacyjnego jest uzyskanie możliwie niskiej ceny konstrukcji. Wymóg ten silnie rzutuje na parametry ruchu manipulatora; dokładność pozycjonowania nie jest zbyt wygórowana (rzędu 1 mm), trudno również mówić o prawdziwej powtarzalności.

Urządzeniem sterującym robotem może być dowolny komputer osobisty wyposażony w standardowe wyjście równoległe lub szeregowe. Niekiedy, wraz z robotem, firmy oferują specjalizowane sterowniki zaopatrzone w odpowiednie przyciski funkcyjne. Jako oprogramowanie podstawowe oferowane są również proste języki programowania robotów, w większości oparte o interpreter języka BASIC lub kompilator języka FORTH. Niektóre systemy są przystosowane do rozbudowy o układy wizyjne i posiadają nawet proste procedury rozpoznawania obrazów. Istnieje kilka typów robotów edukacyjnych, które różnią się zdecydowanie budową od tych omawianych powyżej. Są to jeżdżące mechanizmy-żółwie. Niektóre z nich oprócz sterowanego mechanizmu jezdnego są wyposażone również w ramię prostego manipulatora. Kilka typów robotów edukacyjnych to po prostu plastikowe zabawki o złożonych, mechanicznych lub mechaniczno-elektrycznych układach przeniesienia napędów, sterowane za pomocą dźwigni (joystick). Obserwuje się, że bardziej zaawansowane konstrukcje robotów edukacyjnych (o lepszej powtarzalności ruchu i wyższej precyzji pozycjonowania) znajdują powoli zastosowania również w małych firmach przemysłowych.

Przedstawiony na zdjęciu robot edukacyjny jest efektem pracy kilku osób. Autorem modelu, a następnie prototypu manipulatora jest Paweł Kuzan, absolwent wydziału elektroniki PW. Manipulator w zależności od wykonania posiada 5 stopni swobody (z możliwością przymocowania narzędzia - piórka, wiertarki) lub 4 stopnie swobody i sterowany chwytak. Robot posiada napęd hydrauliczno-elektryczny (pompki o mocy 50W). Do pomiaru położenia wykorzystywane są potencjometry obrotowe.

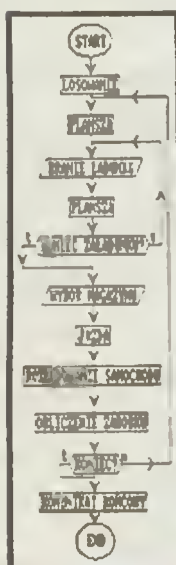
Układ sterowania robota jest autonomicznym mikrokomputerem, zbudowanym z użyciem procesora Z80A, pamięcią EPROM o pojemności 4 kb, pamięcią RAM 8kb, przetwornikiem analogowo-cyfrowym (8 kanałów, — rozdzielczość 8 bitów), układem do transmisji szeregowej oraz układami sterowania silnikiem. Układ sterowania umożliwia nauczenie sekwencji ruchu robota jak i automatyczne odtwarzanie. Ponieważ jednak jest to urządzenie dydaktyczne, przeznaczone dla przyszłych użytkowników robotów a przede wszystkim dla przyszłych konstruktorów, dlatego normalnie pracuje się z robotem w inny sposób. Autonomiczny sterownik łączony jest (złączem szeregowym) z nadrzędnym komputerem (w chwili obecnej gotowe jest oprogramowanie na IBM PC oraz na -MSM — firmy IMPOL-1) i traktowany jest jak inteligentne urządzenie wejścia/wyjścia. Sterownik przyjmuje i wykonuje zdefiniowane makrorozkazy, takie jak: podaj stan robota i urządzeń wejściowych, podaj położenie robota, zapisz sterowanie, zapisz pozycję kroku i wykonaj krok, wczytaj program pracy oraz wczytaj algorytm sterujący. Mając tak bogaty zbiór makrorozkazów pracująca osoba może, w zależności od swoich umiejętności bardziej lub mniej przejmować od komputera władzę nad robotem. W najprostszym przypadku komputer umożliwia zaprogramowanie robota (korzystając z wbudowanego menu), zapamiętanie nauczonego programu na dysku (i oczywiście odczytanie tego programu), wysłanie nauczonego programu do sterownika, a potem wciśnięcie START i już. Natomiast dla prawdziwych fachowców istnieje możliwość zmiany algorytmu sterowania albo przez bezpośrednie sterowanie robocikiem, albo przez przesłanie zmienionego algorytmu pracy do sterownika.

Andrzej Gogolewski

OSIEMNASTOKOŁOWIEC

Cześć Maluchy!

Kubuś Literka, dzielny poszukiwacz przygód i nasz stary przyjaciel został kierowcą wielkiej ciężarówki. Chyba każdy (a może także każda)-z Was marzy o tym, by zasiąść za kierownicą takiego osiemnastokołowca i wyruszyć przed siebie. Okazuje się jednak — przekonał się o tym Kubuś — że życie kierowcy nie składa się z samych tylko przyjemności. Czasem trzeba nieźle się nagłowić, jaki zabrać towar i jaką pojechać trasą, aby przewieźć możliwie najwięcej ładunków.



Zasady gry są bardzo proste. W pięciu miastach znajdują się magazyny z towarem. W każdym z magazynów czekają na transport do poszczególnych miast cztery ładunki. Kubuś rozpoczyna pracę od pierwszego magazynu. Jego ciężarówka zabiera jednorazowo 3 ładunki. Może wyruszyć na trasę tylko siedem razy a równocześnie musi starać się dostarczyć do miejsca przeznaczenia jak najwięcej ładunków. Za każdy z nich otrzymuje zapłatę w wysokości 100 bajtodukatów.

Pisanie naszego programu rozpoczniemy od rozszyfrowania nazw zmiennych i tablic:

pozycja — Określa, w którym mieście znajduje się w tej chwili ciężarówka Kubusia.

zarobek — Suma bajtodukatów, które otrzymał Kubuś za przewiezione ładunki.

magazyn (i, j) — Tablica ta zawiera informacje, jakie ładunki czekają na Kubusia w poszczególnych miastach. Parametr **i** określa miasto a **j** numer ładunku. Jeśli np. magazyn (3,1) = 4 znaczy to, że w mieście 3 pierwszy ładunek czeka na przewiezienie do miasta 4. Wartość 0 oznacza, że ładunku już nie ma.

ładunek (i) — Ta tablica zawiera informacje o ładunku umieszczonym na ciężarówce. Może także zawierać zera, które oznaczają puste miejsca.

Teraz możemy już prześledzić działanie naszego programu. Z pewnością przyda nam się schemat blokowy. Pierwszą czynnością po rozpoczęciu programu oraz deklaracji zmiennych i tablic (linie 10–40) jest wylosowanie początkowego rozmieszczenia ładunków (linie 100–150). Za każdym razem następuje sprawdzenie, czy przypadkiem w którymś z magazynów nie został umieszczony ładunek przeznaczony właśnie do tego magazynu (linia 130).

Po zakończeniu losowania rozpoczyna się właściwa część gry wykonywana w pętli siedmiokrotnie (linie 160–440). Na początek wyświetlana jest na ekranie plansza informująca o stanie gry. W naszym programie planszę drukuje osobny podprogram (linie 1000–1310). Następna czynność to załadunek samochodu (linie 180–290). Nie jest to proste. Komputer musi wiedzieć, ile pustych miejsc jest na samochodzie Kubusia i w zależności od tego, tyle razy pytać o ładunek. Musi także wiedzieć, czy żądany przez grającego ładunek jest w magazynie i aktualizować dane o zawartości magazynów. Jeśli grający nie chce lub nie może zapełnić całego samochodu, na pytanie o ładunek odpowiada wprowadzając wartość 0. Po każdej odpowiedzi wyświetlana jest nowa plansza.

Samochód załadowany, pozostaje tylko decyzja, do którego miasta jedziemy (linie 300–320) i ruszamy w drogę. Krótki program przedstawiający jazdę ciężarówki (linie 330–400) z pewnością wiele zyska, gdy zmienicie go nieco wykorzystując indywidualne możliwości



Waszych komputerów, np. definiowanie znaków graficznych, czy drukowanie w określonym miejscu ekranu. Będzie to pierwsza część Waszego dzisiejszego zadania domowego.

Po dojechaniu na miejsce następuje rozładowanie ciężarówki i obliczenie zarobków Kubusia (linie 410–430). Tu kończy się pętla, a więc powracamy na jej początek, chyba że przebiegamy ją już po raz siódmy. W takim przypadku komputer przechodzi do wyświetlenia komunikatu końcowego i program się kończy.

A ja mam dla Was jeszcze jedną propozycję. Spróbujcie zmienić nasz program w taki sposób, aby można było wiele razy rozpoczynać grę od tej samej sytuacji. W tej chwili, za każdym razem jest ona losowana od nowa.

Powodzenia i... szerokiej drogi życzy Wam

Romek

```
9 REM * deklaracja tablic i zmiennych *
10 LET pozycja=1
20 LET zarobek=0
30 DIM magazyn(5,4)
40 DIM ladunek(3)
99 REM ****** losowanie ladunku ******
100  FOR i=1 TO 5
110    FOR j=1 TO 4
120     LET magazyn(i,j)=INT(RND(1)*5)+1
130     IF magazyn(i,j)=i THEN GOTO 120
140    NEXT j
150  NEXT i
159 REM **** glowna petla programu ****
160  FOR k=1 TO 7
170   GOSUB 1000
179 REM ******* branie ladunku *******
180    FOR i=1 TO 3
190     IF ladunek(i)<>0 THEN GOTO 290
200     PRINT "co zabierasz";
210     INPUT ladunek(i)
220     IF ladunek(i)=0 THEN GOTO 280
230      FOR j=1 TO 4
240       IF magazyn(pozycja,j)=ladunek(
i) THEN LET magazyn(pozycja,j)=0:GOTO 28
0
250      NEXT j
260     LET ladunek(i)=0
270     GOTO 190
280     GOSUB 1000
290    NEXT i
299 REM ******* wybor magazynu *******
300   PRINT "dokad jedziesz";
310   INPUT pozycja
320   IF pozycja>5 OR pozycja<1 OR pozyc
ja<>INT(pozycja) THEN GOTO 300
329 REM *********** jazda ************
330    FOR i=1 TO 35
340     CLS
350      FOR j=1 TO i
360       PRINT " ";
370      NEXT j
380     PRINT "***n"
390       FOR t=1 TO 100:NEXT t
400    NEXT i
409 REM **** rozladunek i zarobek ****
410    FOR i=1 TO 3
420     IF ladunek(i)=pozycja THEN LET l
adunek(i)=0:LET zarobek=zarobek+100
430    NEXT i
440  NEXT k
450 CLS
459 REM ****** komunikat koncowy ******
460 PRINT "Zarobiles"; zarobek; "bajtodu
katow."
470 END
999 REM *********** plansza ***********
1000 CLS
1010 PRINT "-------------"
1020 PRINT "! magazyn";pozycja;"!"
1030 PRINT "-------------"
1040 PRINT
1050 PRINT "          -  !";
1060  FOR m=1 TO 3
1070   IF ladunek(m)<>0 THEN PRINT "---!
";
1080  NEXT m
1090 PRINT
1100 PRINT "         /H! !";
1110  FOR m=1 TO 3
1120   IF ladunek(m)<>0 THEN PRINT ladun
ek(m);"!";
1130  NEXT m
1140 PRINT
1150 PRINT "       E U! ============="
1160 PRINT "       [-o--oo-     -oo-"
1170 PRINT
1180  FOR m=1 TO 5
1190   PRINT "              ";
1200    FOR n=1 TO 4
1210     IF magazyn(m,n)<>0 THEN PRINT "
----";
1220    NEXT n
1230   PRINT
1240   PRINT "magazyn";m;"-!";
1250    FOR n=1 TO 4
1260     IF magazyn(m,n)<>0 THEN PRINT m
agazyn(m,n);"!";
1270    NEXT n
1280   PRINT
1290 PRINT"               ----------------"
1300  NEXT m
1310 RETURN
```

CENY NA DZIEŃ 28.02.88

	GIEŁDA BAJTKA (tys. zł)	PEWEX BALTONA (USD)	WLK BRYTANIA (£)
SINCLAIR			
ZX 81	50	—	30
ZX Spectrum 48 KB	100	115	60
ZX Spectrum Plus	155	—	70
ZX Spectrum 128 + 2	280	—	100
Drukarka SEIKOSHA GP 50S	110	—	40
TIMEX 2048	160	146	—
Joystick	9–11	—	10
COMMODORE			
C-64	245	219	165
C-128	315	299	220
C-128D	—	—	310
Amiga	1,2 mln	—	450
Magnetofon 1531	45	48	25
Stacja dyskietek 1541	260	—	170
Stacja dyskietek 1571	310	299	185
Drukarka GP-500	210	—	—
Dyskietki 5 1/4 (średnia jakość)	0,9–1,0	3,5	0,4
ATARI			
65 XE	175	125	65
130 XE	260	199	105
Stacja dyskietek 1050	220	187	110
Drukarka 1029	250	199	—
ATARI 520 STM st. dysk. 0.5Mb	1,0 mln	798	360
AMSTRAD			
464 z [illegible] monochrom.	[illegible]	—	160
612[illegible] z monit. [illegible]	4[illegible]	—	230
6128 z [illegible] k[illegible]	[illegible]	—	[illegible]
Dys[illegible] 3	6	—	3
PC 1512 SD MM	300	—	[illegible]
PC 15[illegible] SDCM	[illegible]	—	[illegible]
PCW 8[illegible]	[illegible]	—	[illegible]

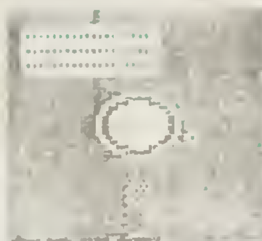
HOME MADE?

Zainteresowanie oprogramowaniem i literaturą fachową nie maleje. Ustaliła się też grupa osób posiadających znakomite rozeznanie potrzeb i wszelkie nowości w polskojęzycznej wersji natychmiast pojawiają się na giełdzie. I chociaż „dorosłe" wydawnictwa proponują coraz ciekawsze pozycje książkowe, ciągle jeszcze nie są one konkurencyjne dla odbitek kserograficznych mizernej jakości, sprzedawanych po 25 zł za stronę.

Miejmy nadzieję, że nowa dziedzina zastosowania techniki komputerowej, czyli Desk Top Publishing zrewolucjonizuje w niedalekiej przyszłości także polski rynek wydawniczy. Posiadając mikrokomputer klasy PC i drukarkę laserową, będzie można nawet na własnym biurku opracować i wydrukować każdą publikację, jakością nie ustępującą profesjonalnemu składowi drukarskiemu.

Póki co, Wydawnictwo Naukowo-Techniczne korzysta ze składu IBM, dzięki czemu cykl wydawniczy skraca do 3-4 miesięcy.

(gr)



INDYWIDUALNY BANK DANYCH

Grzegorz Dudek — uczeń szkoły ponadpodstawowej, lat 15. Posiada mikrokomputer Commodore 16/116. Zainteresowania: informatyka i elektronika. Proponuje wymianę gier oraz programów użytkowych. Adres: 44-100 Gliwice, ul. Lotników 87/29.

Przemysław Sadoch, lat 13. Posiada ATARI 65 XE, joystick, magnetofon XC-12. Zainteresowania: informatyka oraz matematyka. Oprogramowanie: programy matematyczne, muzyczne oraz gry. Proponuje wymianę oprogramowania oraz doświadczeń. Adres: 64-920 Piła, P. Zwycięstwa 10/11

Rafał Okarma, uczeń lat 15. Posiada komputer IBM PC wraz z osprzętem firmowym. Oprogramowanie: programy użytkowe, prezentacyjne i gry. Zainteresowania: informatyka, programowanie. Pragnie nawiązać kontakt z posiadaczami komputerów tej marki, w celu wymiany doświadczeń, literatury i oprogramowania. Adres: 33-171 Pleśna 284, Tarnów

Mariusz Ferdyn, uczeń lat 14. Posiada mikrokomputer Commodore 64, magnetofon TURBO DATA. Proponuje wymianę programów użytkowych oraz doświadczeń. Adres: 03-977 Warszawa, ul. Egipska 5 m 14

Rafał Gałązka, uczeń 12 lat. Posiada mikrokomputer Atari 800 XL, stację dysków 1050. Oprogramowanie: gry i programy użytkowe. Adres: 97-300 Piotrków Trybunalski, ul. Kostromska 66 bl.22 m 10.

Piotr Bilski, lat 26, lekarz. Mikrokomputer AMIGA 500, monitor 1084, drukarka Star NL-10, stacja dysków 3,5". Pragnie nawiązać kontakt z użytkownikami Amigi w celu wymiany doświadczeń, oprogramowania oraz literatury. Adres: 70-836 Szczecin, ul. Węgierska 5.

Karol Augustyn, lat 18. Posiada mikrokomputer Amstrad CPC-464. Pragnie nawiązać korespondencję w celu wymiany doświadczeń oraz oprogramowania. Adres: 15-082 Białystok, ul. Nowotki 13/2 m 59

Ireneusz Niezgoda, lat 15. Posiada mikrokomputer C-Plus 4. Proponuje wymianę oprogramowania i doświadczeń. Odpowie na każdy list za załączeniem znaczka pocztowego. Adres: 05-400 Otwock, ul. Batorego 36 m 18

Marcin Hawryłczup, lat 16. Jest posiadaczem ZX Spectrum 48 KB oraz Commodore 64. Ma dostęp do najnowszego oprogramowania na wyżej wymienione typy mikrokomputerów. Interesuje się informatyką i muzyką. Z posiadaczami tych mikrokomputerów proponuje wymianę oprogramowania oraz doświadczeń. Adres: 03-580 Wa-wa ul. Zamiejska 9 m 19.

Jacek Cieślewicz, uczeń lat 15. Posiada Atari 65 XE oraz magnetofon XC-12. Oprogramowanie: gry i programy użytkowe. Zainteresowania: elektronika, informatyka oraz wędkarstwo. Adres: 62-200 Gniezno, Os. Wł.Łokietka 14 B/15.

Fryderyk Badura, lat 12. Posiada mikrokomputer ZX Spectrum. Proponuje wymianę gier oraz programów użytkowych. Adres: 44-200 Rybnik, ul. Dworcowa 5/41

Drogi Bajtku!

[...] Mój tata kupił komputer o nazwie COMMODORE 2001 PERSONAL ELECTRONIC TRANSLATOR. Podobno jest to dobry komputer ale do biura, do jakichś obliczeń. [...] Pytałem w Bytomiu czy mógłbym dostać programy na kasetach magnetofonowych lecz nic z tego. [...]

Zagubiony RAFAŁ

Komputer o którym mowa zwany jest w skrócie PET i był swoistym hitem firmy Commodore pod koniec lat 70-tych. Był to jeden z pierwszych komputerów przeznaczonych właśnie do biur i zakładów pracy. Jego następcami były Commodore VIC-20, Commodore B128 oraz C-64. Programy do tego komputera są rzeczywiście trudnodostępne jako, ze sprzęt ten został już wycofany z produkcji. Jedynym źródłem mogą być stare czasopisma komputerowe (amerykańskie) z lat 80—83, poza tym giełda, choć obawiam się, ze ich zdobycie będzie dość kłopotliwe. PET nie jest kompatybilny z żadnym innym „dzisiejszym" komputerem Commodore.

Jesteśmy wraz z kolegą początkującymi użytkownikami C-128. Chcielibyśmy dowiedzieć się w jaki najprostszy sposób można połączyć te dwa komputery bez pośrednictwa linii telefonicznej, tak aby możliwe było porozumiewanie się między sobą (mieszkamy oddaleni od siebie o 7 pięter w pionie). Chodzi nam o to, aby sygnał emitowany przez jeden komputer był odbierany zarówno na telewizorze nadającego jak i odbierającego.

Paweł Kamiński, Dariusz Kamiński WARSZAWA

Połączenie takie w zasadzie jest niemożliwe. Po pierwsze w parometrowej długości odcinku przewodu będą się indukowały różnego rodzaju zakłócenia, które w ekstremalnych przypadkach mogą doprowadzić do spalenia układów transmisyjnych i procesora. Po drugie jakiekolwiek uszkodzenie samego przewodu MUSI doprowadzić do spalenia bardzo drogich układów CIA. Połączenie takie może być natomiast realizowane bez przeszkód za pomocą modemów i programu teletransmisyjnego, obojętnie czy będzie to się odbywało poprzez zwykły przewód czy przez linię telefoniczną. W tym ostatnim wypadku wymagana jest jednak uprzednia zgoda odpowiedniego Urzędu Telekomunikacyjnego oraz homologacja modemów.

Mam kilka pytań związanych z mikrokomputerem ZX Spectrum:
1. **Czy istnieje syntezer mowy na ZX Spectrum?**
2. **Gdzie można dostać w Polsce papier do drukarek?**
3. **Jaka gra na ZX Spectrum najlepiej symuluje jazdę samochodem?**
4. **Czy są dwa różne programy o nazwie „GREAT ESCAPE"?**
5. **Jak zmienić kolor rysowanych tinti, punktów w programie ART STUDIO?**
6. **Czy w programie ART STUDIO są opcje rysujące wycinki koła lub elipsy?**
7. **Jaki program na ZX Spectrum pozwala na uzyskanie więcej niż 32 znaków w wierszu?**
8. **Jaki zakład w Polsce rozszerza pamięć RAM ZX Spectrum?**
9. **Gdzie można kupić lub wypożyczyć opisywany na waszych łamach kompilator Pascala?**
10. **Jak można przetłumaczyć program napisany w Basicu nad kod maszynowy?**

**Arkadiusz Socha
ul. J. Kraszewskiego 22/1
57-402 Nowa Ruda**

Oto odpowiedzi na Twoje pytania:
1. Program pozwalający syntezować dźwięki zbliżone do ludzkiej mowy na ZX Spectrum nazywa się LMOWA. „Wypowiedzenie" danego słowa możliwe jest poprzez LPRINT „słowo"
2. Papier do drukarek rozprowadzają firmy zajmujące się sprzedażą komputerów i sprzętu peryferyjnego, głównie drukarek. Oprócz tego papier pojawia się czasem w CSH oraz na giełdach sprzętu komputerowego.
3. Sporo jest gier symulacyjnych, na czoło w symulacjach samochodów wysuwają się TURBO ESPRIT firmy Durell, POLE POSITION firmy Namco oraz FORMULA 1 firmy Mastertronic.
4. Tak. Jeden z nich to Monty's Great Escape firmy Gremlin Graphics z 1985 roku, drugi to opisywany w Bajtku Great Escape firmy Denton Designs z 1986 roku.
5. Po wybraniu z menu opcji ATTR otwiera się okno pozwalające na ustawienie koloru atramentu, tła, ramki oraz jasności, inwersji i nakładania. Kolor wybieramy wskazując strzałką odpowiedni kwadrat.
6. Spośród krzywych stożkowych ART STUDIO może rysować jedynie okręgi. Uzyskanie elipsy możliwe jest przez zmianę skali okna obejmującego wcześniej narysowany okrąg.
7. Chcąc umieścić 64 znaki w wierszu bezpośrednio w języku Basic, użyj programu „64 kolumny tekstu" z Bajtka 7/86. Istnieją programy (głównie edytory tekstów) pozwalające pisać i drukować w podobnej gęstości. Niektóre z nich to Poltasword, The Writer, Star Map.
8. Nie mogę podać Ci adresów firm rozszerzających pamięć ZX Spectrum — przejrzyj dokładnie ogłoszenia w Bajtku i prasie codziennej oraz w innych czasopismach informatycznych.
9. Nie ma w Polsce legalnego źródła uzyskania jakiegokolwiek programu na Spectrum (z wyjątkiem, oczywiście, rodzimych wydawnictw typu KAW). Jedyną możliwością są giełdy komputerowe i kramiki z programami.
10. Istnieje kilka kompilatorów tłumaczących program w Basic-u na pseudo kod maszynowy, jak na przykład FP 48K, BLAST czy TOBOS. Skompilowany program działa około 50 razy szybciej. A czy nie lepiej napisać go od razu w języku maszynowym przy użyciu assemblera np. GENS 3M21?

Przepisałem program „Turbo-Copy" z „Bajtka-Atari". Po uruchomieniu przez RUN pojawia się raport „ERROR- BAT LINE 40". Sprawdzałem, linia 40 jest przepisana prawidłowo. Proszę o podanie prawidłowego brzmienia linii 40.

Tomasz Rowicki

Podobnych listów nadeszło do redakcji wiele, odebraliśmy także liczne telefony. Odpowiedź jest zawsze taka sama: w programie NIE MA błędów. Zamieszczane przez nas programy są testowane i z działającego programu wykonywany jest bisling, który z kolei jest fotografowany i drukowany z otrzymanej kliszy. Na tym etapie nie istnieją żadne możliwości powstania błędów. Natomiast podczas przepisywania programu bardzo łatwo pomylić się i wpisać nieprawidłową liczbę. Zwykle takie błędy są rozpoznawane przez sam program, który wskazuje wtedy linię z błędem. Często jednak przez nieuwagę zamiast cyfry „0" wpisywana jest litera „O" lub kropka zamiast przecinka. Komputer reaguje wtedy wskazaniem błędu nr 8 w linii zawierającej instrukcję READ lub błędu 3 w linii z instrukcją POKE. Aby uniknąć takich przypadków we wszystkich klanach zamieszczone zostały programy kontrolujące poprawność wspisywanego programu przez obliczenie kodu linii. Kody te są podawane obok linii programu. Umożliwia to stwierdzenie poprawności linii przez porównanie otrzymanego kodu z kodem podanym w „Bajtku". Programy kontrolne będą publikowane we wszystkich numerach specjalnych oraz w numerach styczniowych normalnego „Bajtka".

Od niedawna posiadam Atari 800XL. Przez ten krótki czas zorientowałem się już, jak wiele niedogodności niesie ze sobą programowanie w języku Atari Basic. Taśmowe programy z językami są trudno dostępne i wgrywają się zbyt długo, na stację dysków zaś mnie nie stać. W związku z tym proszę o odpowiedzi na następujące pytania:
1. **Czy istnieją cartridge z językami programowania? Z jakimi?**
2. **Jaka jest cena jednego cartridge'a?**
3. **Czy do cartridge'a jest dołączana jakaś literatura?**
4. **Czy cartridge można nabyć w Pewexie? Czy literatura jest wówczas w języku polskim?**
5. **Które z dostępnych na cartridge'ach języków są godne polecenia!**

(nazwisko i adres do wiadomości redakcji)

Wiele języków dostępnych na Atari ma wersje na cartridge'ach. Najważniejsze z nich to: Basic XL, Basic XE, Action!, LOGO, Macroassembler 65, Assembler Editor. Pierwsze dwa nawet nie mają innych wersji. Ceny cartridge'ów są różne — w Pewexie kosztują one około 20$, a w prywatnych i polonijnych firmach oraz na giełdzie około 25—40 tys. zł. W Pewexie cartridge sprzedawane są wraz z oryginalnymi instrukcjami w języku angielskim, natomiast dostępne za złotówki mają zwykle tłumaczenia tych instrukcji (nie zawsze dobrej jakości). Trudno jest polecać któryś z tych języków. Wybór jest zależny od umiejętności programowania, rodzaju programu oraz od upodobań programisty.

JAK REKLAMOWAĆ SIĘ W BAJTKU?

Reklamy przyjmuje Młodzieżowa Agencja Wydawnicza (Redakcja Wydawnictw Poradniczych i Reklamy), 04-028 Warszawa, Al. Stanów Zjednoczonych 53, pokój 313, tel. 10-56-82.
Cena reklamy biało-czarnej wynosi 300 zł za 1 cm^2. Do ceny podstawowej dolicza się 30% za dodatkowy kolor i 100% w przypadku reklamy wielobarwnej. Ogłoszenie drobne kosztuje 200 zł za jedno słowo.

TELEFONICZNE SZALEŃSTWO

(Korespondencja z Japonii)



Japończyk może nie mieć przy sobie drobnych monet 100 lub 10 yenowych. Zawsze jednak ma kartę telefoniczną. Tuż pod otworem wrzutowym znajduje się monitorek, który wyświetla aktualny stan posiadania.

Japończycy tego urządzenia nie wynaleźli. Z pewnością jednak doprowadzili do perfekcji umiejętność posługiwania się nim. Aparat telefoniczny jest w tym kraju sprzętem tak nieodzownym, jak dla nas sztućce do spożywania posiłków. Nic więc w tym dziwnego, że starają się udoskonalać go jak się tylko da.

Żółta, zielone, czerwone, niebieskie. Zainstalowane są na ulicach Toyko dosłownie co krok. Pojedyncze lub zgrupowane po kilka i kilkanaście. Wydawałoby się, że w tej sytuacji nie może byc żadnych problemów ze znalezieniem wolnego automatu. Nic bardziej błędnego. Japończycy pokochali telefon i rozmowy prowadzone przezeń tak samo, i tak szybko, jak pokochali wiele rzeczy rodem ze świata zachodniego. Tylko dzięki ilości automatów nie musimy pokonywać kilkuset metrów w poszukiwaniu wolnego. A kiedy już złapiemy za słuchawkę...

Telefony instalowane na ulicach są dwojakiego rodzaju. Miejskie służąca do krótkich trzyminutowych rozmow i międzymiastowe. Owe trzy minuty kosztuję 10 yenow i jest to najtańsza usługa, jaką można w Japonii spotkac. Oczywiście wrzucenie następnej monety przedłuża nam możliwość dialogu o następne trzy minuty. Automat międzymiastowy kosztuje już 100 yenów. Najczęściej spotkać można połączenie tych dwóch systemów. Żądane połączenie otrzymujemy po wrzuceniu odpowiedniej monety.

Specyficzna miłość do rozmów telefonicznych zainspirowała racjonalizatorow. Myśleli, myśleli i wymyślili automat, do którego nie wrzuca się monet. W zasadzie nie rożni się on niczym od normalnego. Połączenie z abonamentem otrzymujemy po włożeniu karty, na której zakodowano odpowiednią ilość tzw. jednostek rozmówniczych. Zasada posługiwania się tym cudeńkiem jest bardzo prosta. Podnosimy słuchawkę. Do otworu wsuwamy kartę telefoniczną i na malutkim monitorze zostaje wyswietlona liczba jednostek, ktora pozostała jeszcze na karcie. Wykręcamy żądany numer i nie pozostaje nic innego jak gadać. W ciągu całej rozmowy możemy obserwowac „uciekające" cyferki. Kiedy liczba zbliża się do zera, to albo kończymy miły dialog, albo wkładamy nową kartę. I tak bez konca. Po odłożeniu słuchawki przenikliwy sygnał przypomina nam, abysmy nie zapomnieli zabrać karty, która „wypływa" właśnie z automatu.

Przy okazji ruszyła cała machina produkująca karty telefoniczne. Do wyboru i do koloru. Zamierzamy ucinać sobie długie pogaduszki kupujemy taką z tysiącem jednostek. Jeżeli nie należymy do wybitnych gadułow wystarczą nam takie z pięćdziesięcioma (najmniejsza ilość), stoma, stu pięćdziesięcioma. I każda w innym, niepowtarzalnym wzorze i kolorze. A że Japończycy uwielbiają gadać przez telefon przekonałem się, kiedy chciałem zadzwonić do Polski i złożyć życzenia noworoczne (a może to zrobić tylko za pomocą automatu na karty ze względu na wysokość opłaty za rozmowę). Czekałem prawie 30 minut, aż szczebiocząca Japoneczka opowie wszystkie ploteczki swojej znajomej.

O tym, jak Japończycy potrafią ułatwić sobie swoje „telefoniczne życie" mogłem przekonać się jeszcze nie jeden raz. Ot, chociaż w czasie pobytu u znajomego. Zadzwonił telefon. Kolega podniósł słuchawkę i rozmawiał ok. pięciu minut. Nagle słyszę „Poczekaj, ktoś do mnie dzwoni" Nacisnął guzik na aparacie i .. rozmawia z kimś innym. Po pewnym czasie powraca do przerwanego dialogu. Kiedy odłożył słuchawkę poprosiłem o wyjaśnienia. Okazało się, że może w każdym momencie przerwać rozmowę i uzyskać nowa połączenie nie przerywając poprzedniego. Mało tego, w ten sam sposób na jednej linii może ze sobą rozmawiać i kilka osób. Ot, taka telefoniczna konferencja.

Największą frajdę sprawiło mi jednak jednoczesne rozmawianie i widzenie mojego rozmówcy. Aparat telefoniczny z podłączonym do niego monitorem nie należy już do nowości. Można go normalnie kupić w sklepie i zainstalować w domu. Podstawowym warunkiem oglądania rozmówcy jest jednak posiadanie przez niego podobnego urządzenia. Nie są one jeszcze tak popularne jak zwykłe, standardowe aparaty. Jedyną więc możliwość zobaczenia rozmowcy na małym ekraniku miałem w sklepie, w którym bez żadnych problemów udało nam się uciąć kilkuminutową rozmowę na odległość trzech metrów. Zabawa była jednak przednia.

Kiedy siedzieliśmy wieczorem w domu znajomych Japończykow i opowiadaliśmy o wrażeniach minionego dnia, nagle rozległ się głos: „Proszę podnieść słuchawkę, proszę podnieść słuchawkę". W ten sposob telefon dawał znać, że ktoś pilnie chce się połączyć z gospodarzami. Pani domu podniosła słuchawkę i wydała okrzyk radości, gdy się dowiedziała, kto dzwoni. A jej mąż krótko podsumował: „Mamy godzinę z głowy". Niewiele się pomylił.

Jacek Wasilewski



Oryginalna karta telefoniczna. Dziurki obok cyferek określających ilość jednostek rozmówniczych informują o aktualnym stanie posiadania.